

**Henry Joutsijoki ja Erkki Mäkinen (toim.)**

**Tietojenkäsittelytieteellisiä tutkielmia  
Syksy 2017**



TAMPEREEN YLIOPISTO

INFORMAATIOTIETEIDEN YKSIKÖN RAPORTTEJA 57/2017

TAMPERE 2017

TAMPEREEN YLIOPISTO  
INFORMAATIOTIETEIDEN YKSIKÖN RAPORTTEJA 57/2017  
JOULUKUU 2017

**Henry Joutsijoki ja Erkki Mäkinen (toim.)**

**Tietojenkäsittelytieteellisiä tutkielmia  
Syksy 2017**

ISBN 978-952-03-0643-4 (pdf)

ISSN-L 1799-8158

ISSN 1799-8158

## Sisällysluettelo

Samuli Anola-Pukkila	
WWW-sovelluspalvelurajapinnat – REST.....	1
Joni Finckenberg	
Sähköinen ostolaskujärjestelmä: Ostolaskujen käsittelyprosessin automaatio.....	30
Emma Hanhiniemi	
Lisätyn todellisuuden sovellusten käyttömahdollisuuksia neurokirurgisissa toimenpiteissä.....	49
Mikko Helin	
Katsaus roskapostisuodattimiin.....	63
Juuso Kuisma	
Informaation visualisoinnista big dataan: katsaus kirjallisuuteen.....	80
Petja Makkonen	
Tekstinsyöttö laajennetussa todellisuudessa.....	102
Otto-Ville Savolainen	
Pörssikauppaa käyvät algoritmit.....	118
Santeri Tuomisto	
IT-hankintojen ongelmat julkisella sektorilla.....	135
Arttu Ylhävuori	
Pelillisen ja leikillisen oppimisen haasteet ja mahdollisuudet.....	158

# WWW-sovelluspalvelurajapinnat - REST

Samuli Anola-Pukkila

## Tiivistelmä

Tutkielman päämääränä on luoda käytäntöpohjainen kirjallisuuskatsaus www-sovelluspalvelurajapintatoteutuksissa käytettävästä REST-arkkitehtuurimallista. Käytäntöpohjaisen näkökulman vahvistamiseksi tutkielma sisältää kirjoittajan luomia käytännön esimerkkejä, joiden tavoitteena on havainnollistaa käsiteltyjen aiheiden toimintaperiaatteita. Tutkielmassa tarkastellaan REST-arkkitehtuurimallin määritelmästä koituvia etuja sekä haittoja käymällä läpi REST:ssä määritellyt rajoitteet ja erittelemällä kunkin rajoitteen päämäärä sekä siitä aiheutuvat seuraukset kokonaisohjelmistoarkkitehtuuriin. Tutkielman loppupuolella käsitellään myös yleisellä tasolla REST-mallia mukailevan eli RESTful:n www-sovelluspalvelurajapinnan toteutusta. Tutkielmassa avataan REST-arkkitehtuurimallin lisäksi muita www-kehitykseen rinnastettavia aiheita, kuten www-sovellusrajapintoja ja HTTP-protokollaa.

**Avainsanat:** REST, HTTP, URI, rajapinta, www-sovelluspalvelu

## 1 Johdanto

WWW-pohjaisten ohjelmistojen suosion myötä automatisoitujen prosessien tarve kasvaa ja kehittyy [Mumbaikar and Padiya, 2013]. Web-sovelluksien automatisoinnin kulmakivenä toimii ohjelmistojen välisen vuorovaikutuksen mahdollistavat *www-sovelluspalvelut* (engl. web service).

Tässä tutkielmassa tutkitaan julkaisuhetkestään tasaisesti suosiotaan kasvattanutta, yhdeksi alan standardiksi www-sovelluspalveluiden toteutuksissa nousutta REST-arkkitehtuurimallia [Liu *et al.*, 2017; Fielding, 2000]. Suosiosta kielii esimerkiksi suurten tai muuten merkittävien organisaatioiden, kuten Googlen ja Wikipedian, tarjoamat julkiset *RESTful-www-rajapinnat* (engl. RESTful Web API) sekä aiheesta löytyvän kirjallisuuden määrä.

Kirjoittajan motiivina toimii yleisluontoinen kiinnostus web-teknologioita kohtaan, www-sovelluspalvelurajapintojen, -arkkitehtuurimallien sekä -protokollien tuntemuksen tuomat hyödyt www-sovellusta kehittävässä työympäristössä sekä REST-arkkitehtuurimallin tämänhetkinen suosio ja ajankohtaisuus.

Lukijalta *ei odoteta* laajaa teoriapohjaa tai käytännön kokemusta aiheesta. Tästä syystä tutkielmassa tullaan käsittelemään www-sovelluspalveluita

ja REST-arkkitehtuurimalliin liittyviä teknologioita sekä konsepteja perustasolta lähtien.

## 2 WWW-sovelluspalvelut

WWW-sovelluspalveluiden päämääränä on mahdollistaa ohjelmistojen välinen vuorovaikutus internetin välityksellä. Termiä web-sovelluspalvelu ei tule siis sekoittaa termin *web-sovellus* (engl. web application) kanssa; web-sovelluksen päämääränä on toimia ihmisen ja koneen välisessä vuorovaikutuksessa, kun taas web-sovelluspalvelun tehtävänä on toimia esimerkiksi kahden web-sovelluksen välillä.

WWW-sovelluspalvelu on pohjimmiltaan semanttisesti tarkkaan määriteltä abstraktio joukosta erityyppisiä tapahtumia, joiden päämääränä on vaikuttaa ennalta määrättyihin resursseihin ennalta määrättyin tavoin [Sheng *et al.*, 2014]. Tämä tapahtumajoukko määritellään käyttämällä standardioitua ohjelmointikieltä. Vuorovaikutus joukkoon kuuluvien tapahtumien kanssa tapahtuu jonkin standardioituneen internet-pohjaisen protokollan välityksellä.

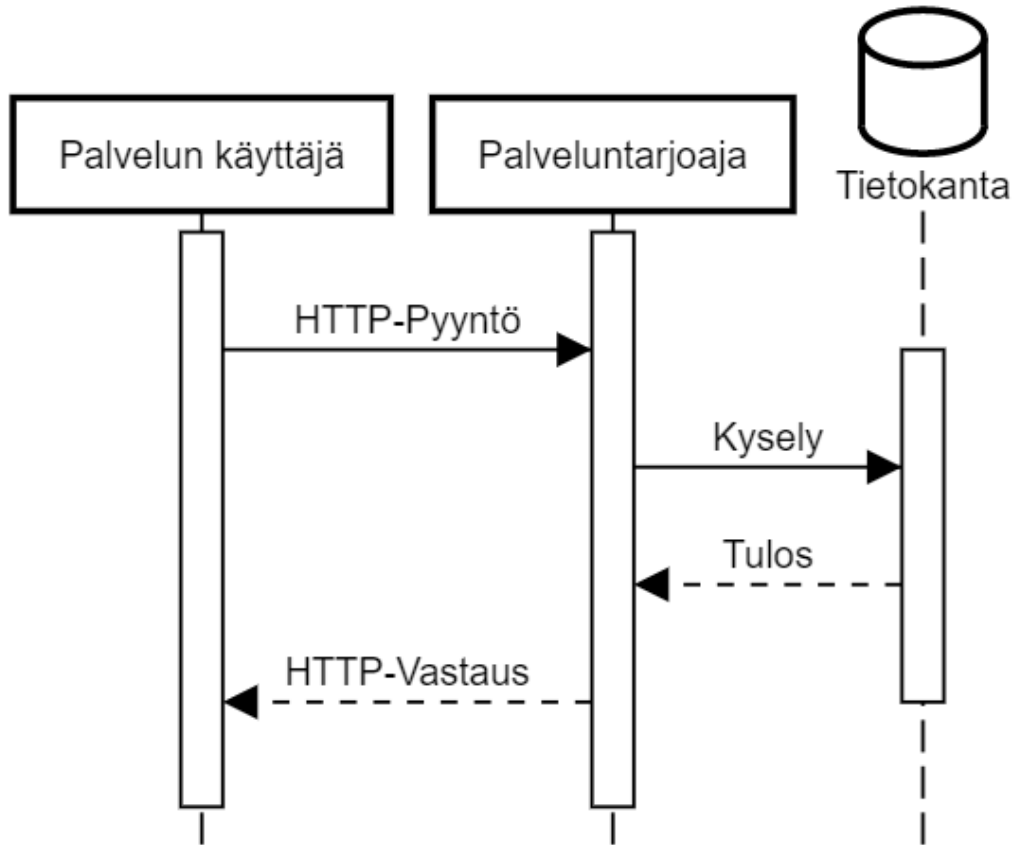
Standardioituneen *tiedonsiirtoprotokollan* (engl. transfer protocol), esimerkiksi HTTP:n, käyttö johtaa siihen, että tapahtumia määrittelevä osapuoli ei määrittele tapahtumia käyttävän osapuolen ominaisuuksia. Tämä tarkoittaa sitä, että määritellyt tapahtumat ovat käytettävissä *aina*, ottamatta kummankaan osapuolen kohdalla huomioon esimerkiksi ohjelmointikieltä, palvelinohjelmistoa tai käyttöjärjestelmää. "Alustariippumattomuus" sekä hyvien suunnittelumenetelmien tuoma *löyhä sidonnaisuus* (engl. loose coupling) palveluja käyttäviin ohjelmistoihin tekee hyvin toteutetusta www-sovelluspalvelusta potentiaalisesti hyvin jatkokehitettävän ja pitkäikäisen kokonaisuuden. [Medjahed, 2004].

WWW-sovelluspalvelussa on kolme osapuolta: *palveluntarjoaja* (engl. service provider), *palvelun käyttäjä* (engl. service requester) ja *palveluhakemisto* (engl. service registry) [Sheng *et al.*, 2014]. Palveluntarjoaja tuottaa ja julkaisee palvelunsa palveluhakemistoon, josta palvelun käyttäjä löytää ja hyödynittää palvelun tarjoamia resursseja. Palveluhakemiston käyttö ei ole pakollista, mutta erityisesti julkisen palvelun löydettävyyden kannalta suositeltavaa.

### 2.1 Käyttö

Eräs yksinkertaisimmista www-sovelluspalveluiden käyttötapauksista on toimia palvelun käyttäjälle, eli esimerkiksi web-sovellukselle resursseja julkaisemaan rajapinnan ja tietokannan yhdistelmänä (kuva 1). Tutkielmassa käytettyjen sekvenssikaavioiden merkintätapa on seuraava: yhtenäinen viiva merkit-

see *pyyntöä* (engl. request) tai tietokantakyselyä, katkoviiva *vastausta* (engl. response) tai tietokantakyselyn tulosta ja pystypalkki sekvenssin osapuolten aktiivisuuden tilaa.

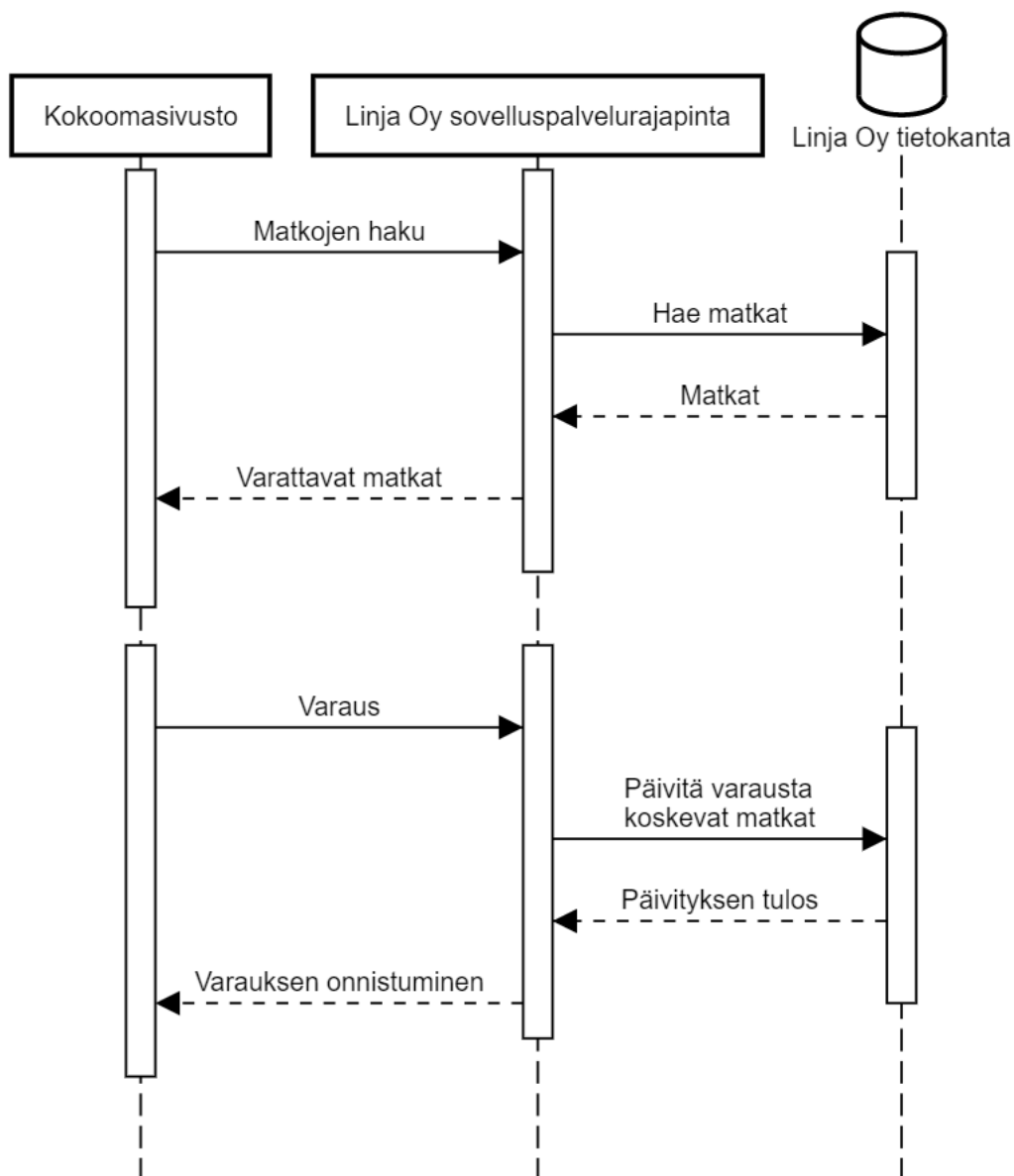


Kuva 1: Yksinkertaisen www-sovelluspalvelun toiminta sekvenssikaaviona.

Yksittäinen taho voi toimia sekä palveluntarjoajana, eli tietokannan ja www-sovelluspalvelun hallinnoijana, että palvelun käyttäjänä. Yleisemmin www-sovelluspalvelun päämääränä on kuitenkin tuoda erinäisten tahojen (esimerkiksi yritysten) sisäisiä prosesseja julkiseen tai rajattuun käyttöön internetin välityksellä [Sheng *et al.*, 2014].

## 2.2 Esimerkkipalvelu - Linja Oy

Kuvitteellisella linja-autoyhtiö *Linja Oy*:llä on verkkosivut, joiden kautta on mahdollista varata lippuja yhtiön tarjoamille matkoille. Alhaisten kävijämäärien vuoksi yhtiö päättää nostaa näkyvyyttään julkaisemalla lipunvarausaikataulunsa sivustolla, joka kokoaa eri linja-autoyhtiöiden aikatauluja yhteen myyden asiakkailleen yhtiöiden tarjoamia matkoja. Tästä sivustosta käytetään jatkossa termiä "*kokoomasivusto*".



Kuva 2: Linja Oy:n matkanvarausrajapinnan matkanvaraussekvenssi.

Yhteistyön mahdollistamiseksi Linja Oy:n täytyy antaa kokoomasivustolle reaaliaikaista tietoa varattavista matkoista sekä ottaa vastaan tietoa kokoomasivustolta varatuista matkoista. Näiden toimintojen automatisointia varten Linja Oy luo rajapinnan vuorovaikuttamaan kokoomasivustolle lähetetyn sekä sieltä saapuvan tiedon kanssa. Koska kokoomasivusto tekee yhteistyötä useampien linja-autoyhtiöiden kanssa, luodaan rajapinta kokoomasivuston esittämien standardien mukaan.

Kuvatuista prosesseista on luotu havainnollistava, esimerkinomainen sekvenssikaavio (kuva 2) on yksinkertaistettu versio todellisesta sekvenssistä;

pyyntöjen muoto on abstraktioitu puhekielelle, ja kaaviosta puuttuu esimerkiksi kannalta epäolennaisia elementtejä, kuten auktorisointi. Puuttuvia asioita käsitellään tarkemmin tutkielman myöhemmissä luvuissa. Sekvenssikaaviossa ei ole myöskään otettu huomioon esimerkiksi kokoomasivuston todennäköisesti käyttämää omaa rajapintaa, vaan oletetaan, että kokoomasivuston pyynnöt tulevat suoraan käyttöliittymästä.

Kuvan 2 kaaviossa kuvataan käyttötapaus, jossa käyttäjä hakee vapaita vuoroja kokoomasivustolta, jonka jälkeen hän suorittaa varauksen Linja Oy:n matkalle. Kuvattu "Matkojen haku" -pyyntö lähetetään kaikille kokoomasivuston kanssa yhteistyötä tekeville linja-autoyhtiöille, jolloin kokoomasivusto kokoaa vastauksina saaduista varattavissa olevista vuoroista koostuvan kokoelman ja esittää tämän käyttäjälle käyttöliittymässä. Kun käyttäjä päättää varata jonkin linja-autoyhtiön tarjoaman matkan, lähetetään tästä tieto kyseisen yhtiön rajapintaan. Tämän jälkeen yhtiön rajapinta päivittää yhtiön tietokannan tilaa halutulla tavalla ja palauttaa kokoomasivustolle tiedon siitä, onnistuiko varaus odotetusti. Tämä mahdollistaa sen, että virhetilanteessa kokoomasivusto kykenee välittämään tiedon epäonnistuneesta varauksesta käyttäjälle.

Kuvassa 2 esitetty *www*-sovelluspalvelurajapinnan ja tietokannan kokonaisuus on siis kuvitteellisen Linja Oy:n tarjoama *www*-sovelluspalvelu. Kuvattu palvelusekvenssi mahdollistaa matkojen automatisoidun, reaaliaikaisen esityksen kokoomasivustolla. Koska palvelu suoritetaan standardioiduilla pyynnöillä käyttäen standardioitua tiedonvälitysprotokollaa, Linja Oy:n tietojen integrointi kokoomasivuston palveluihin ei oikein toteutettuna luo tarvetta muokata aiemmin käytössä olevia järjestelmiä, palvelimia, tietokantoja tai rajapintoja.

### 3 WWW-sovelluspalvelurajapinnat

WWW-sovelluspalvelurajapinta on *www*-sovelluspalvelun julkaistu esitys palvelun tarjoamista tapahtumista. Siinä määritellään julkaistavien tapahtumamethodien käyttämät *viestityypit* (engl. message types) ja *viestinvaihdon mallit* (engl. message exchange patterns) [Brown and Haas, 2004].

WWW-sovelluspalvelurajapinta on siis jokin semanttisesti yhteensopiva kokonaisuus erityyppisiä toimintoja. Työpöytäsovellusrajapintoihin totuneelle konsepti itsessään on tuttu: kirjataan ylös joukko metodeja, joista jokaisella on jokin tehtävä. Tämän tehtävän suoritukseen metodeille voidaan antaa parametreja, ja ne voivat palauttaa paluuarvoja, kuten esimerkiksi tiedon tehtävän onnistumisesta tai metodin avulla haetun muuttujan arvon.



WWW-sovelluspalvelurajapintoihin annettavat parametrit annetaan rajapintoihin tehdyissä, jotakin internet-pohjaista tiedonsiirtoprotokollaa käyttävissä pyynnöissä, ja paluuarvot palautetaan vastauksissa. Tässä tutkielmassa keskitytään HTTP-protokollan käyttöön, joten tästä lähtien tullaan käyttämään termejä *HTTP-pyyntö* (engl. HTTP-request) ja *HTTP-vastaus* (engl. HTTP-response). Tiedonsiirtoprotokollan käyttöön liittyvissä kohdissa viitataan aina HTTP:n käyttöön, jollei toisin mainita. HTTP-protokollaa käsitellään tarkemmin luvussa 4.

### 3.1 Esimerkkirajapinta - Linja Oy

Aiemmin käsittelyssä olleen Linja Oy:n ja kokoomasivuston integraatio tapahtui käytännössä toteuttamalla www-sovelluspalvelurajapinta Linja Oy:n tietokannan, ja kokoomasivuston välille. Yksinkertaistetussa esimerkkitapauksessa kokoomasivuston täytyi suorittaa Linja Oy:n rajapinnassa seuraavia toimintoja:

- noutaa tieto varattavista matkoista
- tallentaa tieto varatusta matkasta.

Jos rajapintaa halutaan kehittää eteenpäin, muokataan Linja Oy:n rajapintaa niin, että se tukee aiemmin kuvattujen toimintojen lisäksi myös seuraavia toimintoja:

- mahdollisuus noutaa tieto varattavista matkoista tietyltä aikaväliltä
- mahdollisuus noutaa tietyn matkan tiedot
- mahdollisuus päivittää varauksen tietoja
- mahdollisuus perua varaus.

Kuvataan nyt kyseisen rajapinnan metodijoukkoa pseudokoodiesityksellä (esimerkkikoodi 1).

```
getTrip(tripId);  
getAvailableTrips();  
getAvailableTrips(startDate, endDate);  
createReservation(reservationInfo);  
updateReservation(reservationId, reservationInfo);  
deleteReservation(reservationId);
```

Esimerkkikoodi 1: Linja Oy rajapinta.

Linja Oy voisi toteuttaa kyseisen rajapinnan sellaisenaan haluamallaan ohjelmointikielellä. Oleellisin ero työpöytärajapintatoteutukseen on se, että

kyseisiä metodeja täytyy pystyä kutsumaan internetin välityksellä. Tämä tapahtuu erillisen tiedonsiirtoprotokollan avulla.

## 4 HTTP

HTTP-protokollaa käsitellään tutkielmassa näennäisen laajasti. Tämä perustellaan HTTP-protokollan ja sen liitännäiskonseptien (esimerkiksi URI ja HTTP-metodien käyttö) olevan *kirjoittajan mielestä* sujuvan REST-www-sovelluspalvelurajapintasuunnittelun sekä -toteutuksen kulmakiviä. Näiden perusteellinen opiskelu saattaa joissain tapauksissa säästää ensimmäistä REST-ful web-rajapintaansa kehittävän ohjelmoijan useiden tuntien refaktoroinnista.

HTTP on sovellustason protokolla hajautetuille, yhdistetyille, hypermediainformaatiojärjestelmille. Se on geneerinen, tilaton protokolla, jota käytetään esimerkiksi tiedonvälitykseen ja hajautettuun hallintaan hyödyntäen HTTP-pyyntöjen metodeja, virhekoodeja ja otsikoita. [Fielding and Reschke, 2014a]. Sovellustasolla viitataan *TCP/IP-mallin* ylimpään kerrokseen [Parziale *et al.*, 2006; Braden, 1989a,b].

Uusin käytössä oleva HTTP-versio on nimeltään *HTTP/2*, jonka erot vanhempaan HTTP/1.1 -versioon liittyvät pääasiassa suorituskykyparannuksiin, ja yksi HTTP/2:n ensisijaisista päämääristä onkin yhteensopivuus (engl. *backwards compatibility*) edeltävien versioiden (HTTP/1.x) kanssa [Belshe *et al.*, 2015]. Tässä tutkielmassa ei keskitytä tarkemmin HTTP-protokollan versioiden eroihin, tai oteta kantaa HTTP/2:n käyttöönoton kannattavuuteen. On kuitenkin huomionarvoista tutkielmassa käsiteltyjen asioiden pätevän kaikilla yleisesti käytössä olevilla protokollan versioilla (HTTP/2, sekä HTTP/1.x).

### 4.1 HTTP-viestien rakenne

Koneellinen kommunikaatio HTTP:n avulla koostuu HTTP-viesteistä. HTTP-viesti voi olla joko johonkin resurssiin kohdistettu HTTP-pyyntö tai kyseisestä resurssista tuleva HTTP-vastaus. Nämä viestit koostuvat *aloitusrivistä* (engl. start-line),  $N \geq 0$  kappaleesta *otsikkorivejä* (engl. request-header) sekä mahdollisesta *viestirungosta* (engl. message-body). Syntaktisesti ainoa ero HTTP-pyyntö- ja HTTP-vastauksella on aloitusrivi, joka on joko *pyyntöriivi* (engl. request-line) tai *tilarivi* (engl. status-line). [Fielding and Reschke, 2014a].

```
POST /post HTTP/1.1
cache-control: no-cache
content-type: text/plain
user-agent: PostmanRuntime/6.4.1
host: postman-echo.com
content-length: 256

Duis posuere augue vel cursus pharetra. In luctus a ex nec pretium.
```

Esimerkkikoodi 2: HTTP-pyyntö.

```
HTTP/1.1 200
status: 200
content-encoding: gzip
content-type: application/json; charset=utf-8
cache-control: no-cache
date: Mon, 27 Nov 2017 16:26:24 GMT
server: nginx
vary: X-HTTP-Method-Override, Accept-Encoding
content-length: 524
connection: keep-alive
{
  "args": {},
  "data": "Duis posuere augue ve.."
}
```

Esimerkkikoodi 3: HTTP-vastaus.

Tarkastellaan HTTP-viestien rakennetta esimerkkien (esimerkkikoodit 2 ja 3) avulla. Kummassakin esimerkkikoodissa ensimmäisenä oleva rivi on aiemmin mainittu aloitusrivi. HTTP-pyyntö kohdalla siihen on merkitty käytetty metodi (POST), resurssin sijainti palvelimella (/post) sekä käytetty HTTP-protokollan versio (HTTP/1.1). HTTP-vastauksen kohdalla aloitusriville on merkitty HTTP-protokollan versio sekä pyynnön käsittelyn tuottama status (200). Aloitusrivin jälkeen tulevat viestien mukana kulkevat otsikkorivit. Näistä riveistä käy ilmi esimerkiksi resurssia ajavan palvelimen juuriosoite (host: postman-echo.com). Otsikkorivien jälkeen tulee otsikkorivien päättymisestä ilmaiseva tyhjä rivi sekä viestirungot.

## 4.2 Metodit

HTTP-pyynnöt lähetetään aina käyttäen jotakin HTTP-protokollaan määriteltä, pyynnön päämäärää kuvaavaa metodia. Tämä päämäärä voi olla esimerkiksi tiedonhaku tai -tallennus. Nämä metodit jakautuvat kolmeen eri kategoriaan, *turvallisiin* (engl. safe methods), *idempotentteihin* (engl. idempotent methods) sekä muihin metodeihin (taulukko 1 ja esimerkkikoodi 4). Kappaleessa käsitellään vain tutkielman näkökulmasta oleellisia metodeja.

Metodi	Turvallinen	Idempotentti
OPTIONS	Kyllä	Kyllä
GET	Kyllä	Kyllä
HEAD	Kyllä	Kyllä
POST	Ei	Ei
PUT	Ei	Kyllä
DELETE	Ei	Kyllä
PATCH	Ei	Ei

Taulukko 1: Yleisten HTTP-metodien turvallisuus ja idempotenttisuus.

```
class Example {  
  
    int global = 0  
  
    safeAndIdempotentMethod() {  
        return global  
    }  
  
    idempotentMethod(value) {  
        global = value  
    }  
  
    notSafeNorIdempotentMethod() {  
        global++  
    }  
}
```

Esimerkkikoodi 4: Havainnollistava pseudokoodiesitys esimerkkiluokasta.

Turvallinen metodi on metodi, jolla ei voida muuttaa kohderesursseja. Tämä tarkoittaa käytännössä metodia, joka on semanttisesti "vain luku"

-tilassa. [Fielding and Reschke, 2014b]. Yleisesti turvallisia metodeja käytetään tiedonhakuun (esimerkkikoodi 4, *safeAndIdempotentMethod*-metodi).

Idempotentilla metodilla voidaan muuttaa resursseja. Idempotentilla metodilla on kuitenkin rajoituksena  $N > 0$  identtisen pyynnön lopputulosten identtisyys. Tämä tarkoittaa sitä, että yhden pyynnön toteutuessa kyseistä pyyntöä seuraavat identtiset pyynnot johtavat aina samaan lopputulokseen kuin alkuperäinen pyyntö. Idempotentin metodin käyttö jättää siis järjestelmän aina samaan tilaan ensimmäistä pyyntöä seuraavien identtisten pyyntöjen määrästä riippumatta. [Fielding and Reschke, 2014b]. Yleisesti idempotentteja metodeja käytetään tiedon hakuun, muokkaamiseen, tai poistamiseen (esimerkkikoodi 4, *idempotentMethod*-metodi).

Muut metodit ovat metodeja, jotka eivät täytä turvallisen tai idempotentin metodin kriteerejä. Tällaisia metodeja käytetään esimerkiksi uuden tiedon luontiin tai tiedon muuttamiseen esimerkkikoodi 4:n *notSafeNorIdempotentMethod*-nimisessä metodissa kuvatulla inkrementoiavalla tavalla.

GET-metodilla haetaan haluttuja resursseja jossakin tarjolla olevassa esitysmuodossa. Se on HTTP:n ensisijainen tiedonhakuun tarkoitettu metodi. [Fielding and Reschke, 2014b].

HEAD-metodi on muuten identtinen GET-metodin kanssa, mutta HEAD-pyyntö HTTP-vastauksessa *ei saa* olla body-osaa eli viestirunkoa [Fielding and Reschke, 2014b]. HEAD-metodia voidaan käyttää esimerkiksi haettaessa resurssiin liittyvää metatietoa.

POST-metodia käytetään pyyntönä, jonka päämääränä on käskä kohderesurssia prosessoimaan pyynnön mukana tulevaa tietoa resurssin omien semanttisten sääntöjen mukaisesti. Yleisesti POST-metodia käytetään tiedon inkrementoivaan lisäämiseen (esimerkiksi viestin lähettäminen keskustelualueelle), tiedon lisäämiseen jo olemassa olevaan, kohderesurssissa sijaitsevaan tietorakenteeseen tai tiedon kuljettamiseen tietojenkäsittelyprosessia varten (esimerkiksi HTML form -elementin käyttö). [Fielding and Reschke, 2014b].

PUT-metodia käytetään tiedon idempotenttiin lisäykseen tai muokkaukseen. Tämä tarkoittaa sitä, että PUT-pyyntö joko luo uuden tai korvaa olemassaolevan resurssin. [Fielding and Reschke, 2014b]. Sekä POST:ia, että PUT:ia voidaan käyttää tiedon luomiseen tai päivittämiseen, eikä niiden välillä ole suurta semanttista eroa. Pääasiallinen ero kehittäjän näkökulmasta on toiminnon idempotenttisuus. Jos päämääränä on idempotentti tietojenkäsittelytapauksena, käytetään PUT:ia ja muutoin käytetään tilanteen mukaan joko POST:ia tai PATCH:ia.

PATCH-metodia käytetään olemassa olevan resurssin osittaiseen muokkaukseen. Tämä eroaa PUT:sta sillä, että PATCH ei ole idempotentti metodi. Käytettäessä PUT-metodia resurssin muokkaukseen, korvataan kohdere-

surssi kokonaan, kun taas PATCH muokkaa resurssia vain osittain. Joskus PUT:n käyttö PATCH-metodin sijaan on suositeltavaa myös näennäisesti osittaisessa muokkauksessa, esimerkiksi silloin, kun PATCH-metodin toteutukseen tarvittava tietomäärä on huomattavasti suurempi kuin kokonaan uuden resurssin luominen PUT:lla. POST-metodia käytetään pääasiassa resursien luontiin, joten PATCH:ia voi yleisesti käyttää tilanteissa, joissa POST:ia käytettäisiin resurssin muokkaukseen. [Dusseault and Snell, 2010].

DELETE-metodia käytetään kohderesurssin poistoon. Fielding ja Reschke [2014b] vertaavat DELETE-metodia UNIX-käyttöjärjestelmän *rm*-komentoon tuoden esiin DELETE-metodin päämäärän poistaa nimenomaan tiettyssä osoitteessa sijaitseva resurssi jättäen aiemmat informaatioassosiaatiot huomiotta.

OPTIONS-metodilla pyydetään tietoja kohderesurssiin kohdistuvista tietojenkäsittelymahdollisuuksista [Fielding and Reschke, 2014b]. Näin voidaan hakea esimerkiksi tarkat ohjeet jonkin resurssin luontia varten.

## 4.3 URI

Uniform Resource Identifier, eli URI, on kompakti merkkijono, joka yksilöi ja paikallistaa joko abstraktin tai fyysisen resurssin. URI voidaan tarkemmin luokitella kuvaamaan joko resurssin sijaintia (URL), nimeä (URN) tai kumpaakin edeltävistä. HTTP:ta käytettäessä pyritään paikantamaan kohderesurssit tietoverkossa, joten metodien kohderesursseista puhuttaessa voidaan yleisesti käyttää myös termiä URL. URI:n oletussyntaksi koostuu neljästä pääkomponentista: skeemasta, auktoriteetista, polusta, ja kyselystä (esimerkkikoodi 5). [Berners-Lee *et al.*, 2005].

<code>&lt;scheme&gt;://&lt;authority&gt;&lt;path&gt;?&lt;query&gt;</code>
---

Esimerkkikoodi 5: Geneerisen URI syntaksin neljä pääkomponenttia.

Skeema viittaa URI:n syntaksia määrittelevään, rekisteröityyn protokollaan, jollainen myös HTTP on [Thaler *et al.*, 2015]. Auktoriteetti koostuu valinnaisesta autentikaatio-osasta, johon kuuluvat käyttäjänimi ja salasana, sekä *isäntäosasta* (engl. host), joka määrittelee HTTP:ta käytettäessä joko rekisteröidyn verkkosivun nimen tai IP-osoitteen eli resurssin juuren. Polku määrittelee resurssin sijainnin suhteessa aiemmin määritettyyn resurssin juureen. Kyselyn avulla voidaan rajata resurssia tai vaikuttaa sen esitystapaan. [Berners-Lee *et al.*, 2005].

HTTP:n oma URI-syntaksin määritelmä (esimerkkikoodi 6) [Fielding and

Reschke, 2014a, 17-18] mukailee yleistä URI:n määritelmää poislukien HTTP-kohtaiset tarkennukset.

`"http:" "://" authority path-abempty [ "?" query ] [ "#" fragment ]`

Esimerkkikoodi 6: HTTP:n URI-syntaksi.

HTTP-URI alkaa protokollan määrittelyllä, eli merkkijonolla "*http*". Tämän jälkeen merkitään *auktoriteettiosa* (authority), jota edeltää kaksinkertainen kenoviiva "//", ja joka päättyy joko kenoviivaan "/", kysymysmerkkiin "?", numeromerkkiin "#" tai URI:n loppuun. Auktoriteettiosa koostuu valinnaisesta autentikaatio-osasta, isäntäosasta (palvelimen juuriosoite), sekä portista (oletuksena 80). Komponentti *path-abempty* viittaa yleisessä URI-syntaksin määrittelyssä määriteltyyn *polku*-osaan, joka alkaa kenoviivalla "/" tai on tyhjä. *Kysely* (query) viittaa URI-kyselyyn ja *kappale* (fragment) viittaa tiettyyn resurssin osaan, esimerkiksi artikkelin alaotsikkoon. [Berners-Lee *et al.*, 2005, 17-25].

`http://10.0.0.30:8080/article?title=http&date=17-06-18#basics`

Esimerkkikoodi 7: HTTP:n URI-syntaksin mukainen URI.

## 5 REST

REST, eli *representational state transfer*, on web-sovellusarkkitehtuurissa laajalti käytössä oleva ohjelmistoarkkitehtuurimalli. Tässä tutkielmassa REST-arkkitehtuurimallia käsitellään vahvasti www-sovelluspalvelurajapintojen kautta, mutta mallin alkuperäinen kuvaus ei suoranaisesti sido REST:iä rajapintoihin, vaan mallin esiintuonut Fielding [2000] kuvaa sitä "*arkkitehtuaaliseksi tyyliksi hajautetuille hypermediajärjestelmille*".

Nykyaikaisessa www-sovellusarkkitehtuurissa käytettynä REST rakentuu URI:lla tunnistettavien *esitysmuotoisten* (engl. representational) tiettyssä *tilassa* (engl. state) olevien resurssien HTTP-protokollalla tehtyjen *siirtojen* (engl. transfer) ympärille. Ne koostuvat erinäisistä arkkitehtuaalisista rajoitteista, joiden pyrkimyksenä on rakentaa skaalautuvia, uudelleenkäytettäviä, löyhästi sidonnaisia, korkeasuorituskykyisiä ja matalalla viiveellä toimivia rajapintoja [Mumbaikar and Padiya, 2013; Rodriguez, 2008; Pautasso, 2014].

Tutkielman aiemmissa luvuissa on käytetty yleistajuista sanaa resurssi kuvaamaan tietoa tai tietolähdettä. REST:ssä resurssi määritellään erikseen:

resurssi on mitä tahansa nimettävissä olevaa tietoa. Tämä tieto voi olla esimerkiksi dokumentti, kuva, merkkijono, kokoelma muita resursseja tai täysin ei-virtuaalinen objekti (kuten ihminen).

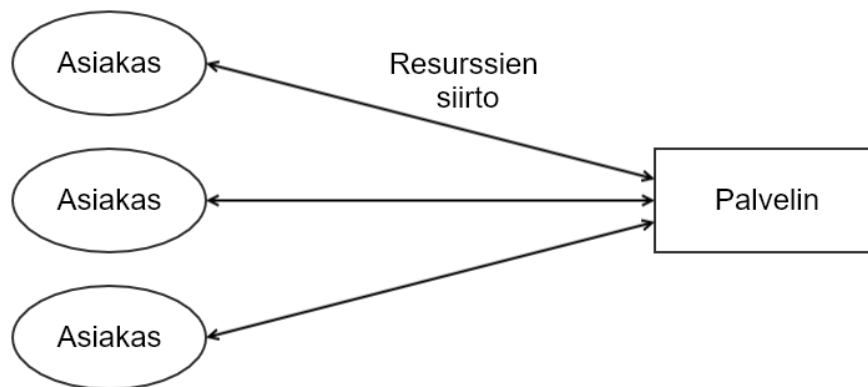
REST-resurssi ei kuitenkaan viittaa itse tietoalkioon, vaan se on viite vastaavista tietoalkioista koostuvaan tietojoukkoon. Tällöin resurssi ei käytännössä viittaa alkioon tai alkiojoukkoon, joka jonakin tiettyinä ajankohtana edustaa viitattavan tietojoukon tilaa. Kuitenkin se samalla viittaa tietojoukon, eli tietoalkion arvoon. [Fielding, 2000; Vinoski, 2008].

Tutkielman seuraavissa luvuissa käsitellään REST:n määritteleviä arkkitehtuaalisia rajoitteita sekä tarkastellaan REST-arkkitehtuurimallilla toteutettavaa www-sovelluspalvelurajapintaa käytännön tasolla.

## 6 REST - Arkkitehtuaaliset rajoitteet

REST-rajapinta-arkkitehtuurimallin rajoitteet rakentuvat ns. *tyhjän tyylin* (engl. null style) päälle. Tyhjä tyyli viittaa järjestelmään, joka ei sisällä rajoitteita järjestelmän eri komponenttien välillä. [Fielding, 2000, 77].

### 6.1 Asiakas-palvelin -arkkitehtuuri

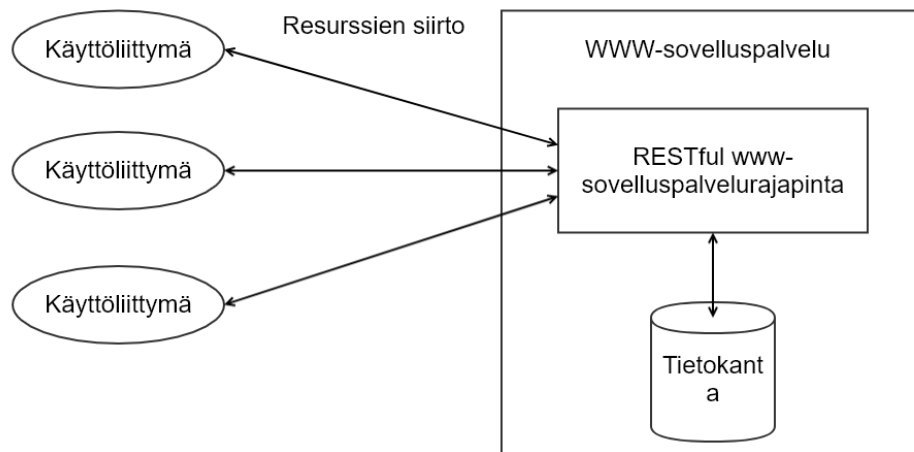


Kuva 3: Asiakas-palvelin -arkkitehtuurimalli.

*Asiakas-palvelin -arkkitehtuurin* (engl. client-server architecture) (kuvat 3 ja 4) päämääränä on erottaa asiakas palvelimesta. Käytännössä tämä tarkoittaa usein käyttöliittymän erottamista kohderesursseja hallinnoivasta osapuolesta eli esimerkiksi www-sovelluspalvelurajapinnasta. Näiden osapuolten erottamisella pyritään parantamaan käyttöliittymän siirrettävyyttä eri alusto-



jen välillä edistään skaalautuvuutta yksinkertaistamalla palvelinkomponenttien toimintaa. Alkuperäisessä määritelmässä mahdollisesti merkittävimpänä seikkana asiakkaan ja palvelimen erotuksessa pidettiin kykyä kehittää kumpaakin osapuolta itsenäisesti. [Fielding, 2000, 78].



Kuva 4: Asiakas-palvelin -arkkitehtuurimalli www-sovelluspalveluna.

## 6.2 Tilattomuus

*Tilattomuus* (engl. statelessness) lisää aiemmin mainittuun asiakas-palvelin -arkkitehtuuriin seuraavan rajoitteen: kommunikaatio näiden kahden entiteetin välillä tulee olla luonteeltaan tilatonta [Fielding, 2000, 78].

Tilattomuus merkitsee jokaisen asiakkaalta palvelimelle lähetetyn pyynnön täytyvän sisältää kaikki tarvittava tieto pyynnön jäsentelyä varten sekä asiakasosapuolen kykenemättömyyttä hyväksikäyttää palvelimelle tallentunutta kontekstuaalista tietoa. Palvelimen ja asiakkaan välille muodostuvaan *istuntoon* (engl. session) liittyvä tieto, kuten käyttäjätiedot, tallennetaan siis aina asiakaspuolelle, ja jokaisen osapuolten välisen vuorovaikutustapahtuman jälkeen asiakas ja palvelin eivät jaa yhteisiä tilatietoja. [Fielding, 2000; Pautasso, 2014].

Tämä rajoite varmistaa sen, että resurssiin kohdistuvat pyynnöt eivät ole riippuvaisia toisistaan [Pautasso, 2014, 33]. Se edistää näkyvyyttä, luotettavuutta sekä skaalautuvuutta. Näkyvyys paranee, sillä järjestelmää monitoroiva osapuoli kykenee havainnoimaan asiakkaan lähettämän pyynnön luonteen tarkastelemalla vain yhtä pyyntötietuetta. Luotettavuus paranee, sillä pyyntöjen tilattomuus helpottaa palautumista osittaisista virhetilanteista

[Fielding, 2000, 79; Waldo *et al.* 1996]. Skaalautuvuus paranee, sillä potentiaalisesti tarpeellisten tilatietojen puuttuessa palvelimella on mahdollisuus vapauttaa resurssejaan nopeasti. Lisäksi tilattomuus yksinkertaistaa palvelintoteutuksen rakennetta, joka helpottaa palvelinohjelmiston täytöntöönpanoa.

Tilattomuudella on myös haittapuolia. Näihin lukeutuu verkkoliikenteen mahdollinen kuormittuminen asiakkaan ja palvelimen välisen kontekstin ollessa mukana jokaisessa palvelimelle tehdyssä pyynnössä, joka nostaa *toistuvan tiedon välityksen* (engl. per-interaction overhead) määrää järjestelmässä. Tämänkaltaiselta tarpeettomalta tiedonvälitykseltä ei voida välttyä, jos osapuolien välistä kontekstia ei saa säilyttää palvelimella. [Fielding, 2000, 79].

Tilan tallentuminen asiakkaan puolelle tarkoittaa myös sitä, ettei palvelinohjelmisto tai sen hallinnoija kykene varmistamaan palvelinohjelmiston tarjoamien resurssien tarkoituksenmukaista käyttöä. [Fielding, 2000, 79]. Tämä voi vaarantaa tiedon eheyden esimerkiksi tapauksessa, jossa palvelinohjelmiston julkaisemat resurssit ovat muuttuneet, mutta logiikkavirheen vuoksi asiakasohjelma ei ole pyytänyt päivitettyjä tietoja palvelimelta näyttäen käyttöliittymässään virheellistä tietoa.

## 6.3 Välimuisti

*HTTP-välimuisti* (engl. HTTP cache) on paikallinen varasto HTTP-pyyntöjen vastauksille ja välimuistin toimintoja hallinnoivalle osajärjestelmälle. Välimuisti varastoi *välimuistikelvollisia* (engl. cacheable) HTTP-vastauksia vähentääkseen vastausaikoja ja verkon kaistanleveyden käyttöä tulevilla vastaavilla HTTP-pyyntöillä. [Fielding *et al.*, 2014, 4].

Välimuistirajoite pyrkiikin kompensoimaan tilattomuusrajoitteesta aiheutuvia mahdollisia suorituskykyhaittoja. Välimuistirajoitteen mukaan palvelimen antaman vastauksen sisältämän tiedon täytyy olla merkitty joko välimuistikelvolliseksi tai välimuistikelvottomaksi. Jos vastaus on välimuistikelvollinen, asiakasohjelma saa oikeuden käyttää jatkossa vastauksen sisältämää tietoa vastaaviin pyyntöihin. [Fielding, 2000, 79]. Välimuistikelvollisuus voidaan määritellä HTTP-viestin otsikossa *cache-control* (esimerkkikoodi 2).

Tiedon välimuistiin tallentamisen etuna on joidenkin vuorovaikutustapahtumien osittainen tai täysinäinen poisto tapahtumaketjusta. Tämä parantaa tehokkuutta, skaalautuvuutta sekä käyttäjän havainnoimaa suorituskykyä laskemalla keskiarvoista viivettä tapahtumaketjujen välillä.

Välimuistiin tallentamista täytyy kuitenkin käyttää harkiten, sillä se saattaa vähentää tiedon luotettavuutta. Ongelma ilmenee, kun vanha, välimuistiin tallennettu tieto eroaa merkittävästi suoraan palvelimelle lähetetyn pyyn-

nön palauttamasta tiedosta. [Fielding, 2000, 80].

## 6.4 Rajapinnan yhdenmukaisuus

Fielding [2000] kuvailee rajapinnan yhdenmukaisuuden olevan keskeisin ero REST:n ja muiden verkkopohjaisten arkkitehtuurimallien välillä. Järjestelmän komponentit yhdistetään toisiinsa yhdenmukaisen rajapinnan kautta, jolloin kaikkien järjestelmässä käsiteltävien resurssien on kuljettava kyseisen rajapinnan läpi. Tämä rajapinta tarjoaa kokoelman pieniä, geneerisiä, funktionaalisesti riittäviä metodeja tukemaan kaikkia mahdollisia vuorovaikutustapahtumia resurssien välillä. [Pautasso, 2014, 33].

Yhdenmukaisen rajapinnan käyttö yksinkertaistaa järjestelmäarkkitehtuuria kokonaisuudessaan, ja parantaa vuorovaikutustapahtumien näkyvyyttä niiden kulkiessa aina saman rajapinnan kautta. Yhdenmukainen rajapinta helpottaa myös komponenttien itsenäistä kehitystä asiakas-palvelin -arkkitehtuurin pohjalta rakennetussa järjestelmässä. [Fielding, 2000, 81].

Yhdenmukaisen rajapinnan huonoksi puoleksi luetaan tehokkuuden heikentyminen eli suorituskyvyn lasku. Tämä johtuu käsiteltävän tiedon standardoidusta muodosta, joka ei ole välttämättä optimaalisin jokaiselle rajapintaa käyttävälle sovellustoteutukselle. [Fielding, 2000, 82].

WWW-sovelluspalveluiden ja HTTP-protokollan näkökulmasta aiemmin kuvattu yhdenmukainen rajapinta koostuu HTTP-metodeista, jotka voidaan kohdentaa käsittelemään tiettyä resurssia yksilöimällä kyseinen resurssi URI:lla [Pautasso, 2014, 33].

REST määrittelee yhdenmukaiselle rajapinnalle lisärajoitteita, joiden mukaan kyseinen komponentti tulee toteuttaa: *resurssien tunnistus* (engl. identification of resources), *resurssien manipulointi esitysten kautta* (engl. resource manipulation through representations), *viestien itseselitteisyys* (engl. self-descriptive message) ja *hypermedia sovelluksen tilan ajurina* (engl. hypermedia as the engine of application state, tai *HATEOAS*) [Fielding, 2000, 82].

Resurssien tunnistus www-pohjaisissa RESTful-rajapinnoissa tapahtuu URI:n avulla [Pautasso, 2009, 854]. REST-malli kannattaa intuitiivisten ja helppolukuisten URI:en käyttöä [Rodriguez, 2008; Pautasso *et al.*, 2008]. REST-mallin mukaisia URI:ja käsitellään tarkemmin kohdassa 7.1.

*Esitys* tarkoittaa REST:n kontekstissa mitä tahansa hyödyllistä informaatiota jonkin resurssin tilasta. Resurssilla voi olla useita eri esityksiä tai esitystapoja. [Feng *et al.*, 2009]. Nämä esitystavat ovat käytännössä resurssien *mediatyyppejä* (engl. media type, aiemmin MIME-tyyppi, esimerkiksi *text/plain* tai *image/jpeg*) esityksiä sekä mahdollisia metatietoja kyseisistä

esityksistä [Freed and Borenstein, 1996; Alarcón and Wilde, 2010]. Mediatyyppi eli resurssin esitysmuoto määrittää HTTP-viestin otsikossa *content-type* (esimerkkikoodit 2 ja 3).

Itseselitteisellä viestillä tarkoitetaan sitä, että jokainen viesti sisältää kaikki tarvittavat tiedot ja metatiedot halutun tapahtuman tarkkaan toteutukseen. Tällaisia tietoja ovat esimerkiksi aiemmin mainitun mediatyyppin tai jonkin resurssin, kuten artikkelin esityskielen määrittely. [Pautasso, 2014, 33].

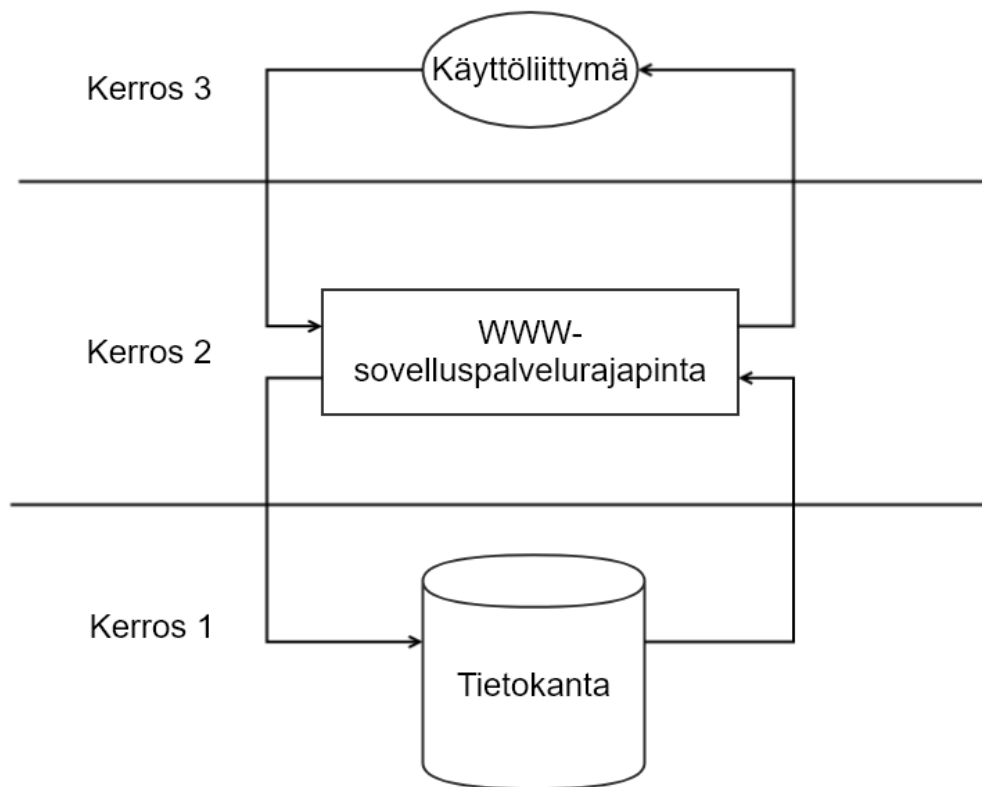
Hypermedia sovelluksen tilan ajurina rajoite pyrkii siihen, että RESTful API:n resurssien yhteyksiä sekä mahdollisia tilanmuutoksia mallinnettaisiin eräänlaisella hypermediasta koostuvalla viitteiden verkolla. Tämä mahdollistaa sen, ettei REST-palvelu toimi pelkkänä *päätepisteiden* (engl. endpoint) kokoelmana, vaan se voi tarjota pienen määrän URI:lla kohdennettuja resursseja, jotka toimivat *sisääntulopisteinä* (engl. entry-point) suuremmalle palvelukokonaisuudelle. [Alarcon *et al.*, 2010].

Hypermedia auttaa siis hajautetussa resurssien löytämisessä (engl. decentralized resource discovery), dynaamisessa löytämisessä (engl. dynamic discovery) ja vuorovaikutusprotokollien määrittelyssä palveluiden välillä [Pautasso, 2014].

Hypermedian käyttöä ei kuitenkaan aina huomioida REST-mallia mukaillevan rajapinnan toteutuksissa. Fielding [2008], Vinoski [2008] ja Pautasso [2014] nostavat esiin ongelman hypermedian ja aiemmin kuvatun palvelukokonaisuuden verkotuksen vähäisestä käytöstä niin kutsutuissa RESTful-rajapinnoissa, ja arvioivat HATEOAS:n olevan yksi REST:n useimmin rikotuista rajoitteista. Rajoitteen käyttöä havainnollistetaan kohdassa 7.2.

## 6.5 Järjestelmän kerroksittaisuus

Kerroksittainen järjestelmä (kuva 5) koostuu kerroksista, jotka koostuvat komponenteista, joita ovat esimerkiksi käyttöliittymä ja www-sovellusrajapinta. Kerrokseen kuuluva komponentti kykenee vuorovaikuttamaan vain välittömästi vierekkäisiin kerroksiin kuuluvien komponenttien kanssa [Fielding, 2000, 82-83]. Tämä merkitsee sitä, ettei kerroksessa oleva komponentti ole tietoinen muiden kuin sen välittömässä läheisyydessä olevien kerrosten olemassaolosta, jolloin esimerkiksi käyttöliittymäkomponentti ei voi tietää, onko se yhteydessä suoraan palvelun sisältämään palvelimeen vai johonkin välittäjäpalveluun.



Kuva 5: Kerroksittainen järjestelmä.

Rajoittamalla komponenttien näkyvyyttä saadaan rajoitettua järjestelmän potentiaalista monimutkaisuutta. Lisäämällä kerroksia voidaan yksinkertaistaa monimutkaisten ongelmien käsittelyä pilkkomalla ne useaan abstraktioituun ohjelmiston tasoon [Garlan and Shaw, 1993, 11]. Kerroksia voidaan hyödyntää vanhentuneiden palveluiden kapseloinnissa ja uusien palveluiden suojaamisessa vanhentuneilta asiakasohjelmilta.

Välittäjäpalveluilla voidaan tasapainottaa esimerkiksi tietokantakuormitusta jakamalla pyynnöt useiden prosessointiyksiköiden ja verkkojen välille. Niiden avulla pystytään esimerkiksi autentikoimaan käyttäjiä ja parantamaan tietoturvaa lisäämällä palomuurien vaatimia turvallisuutta koskevia toimia organisaation verkon rajalle. Koska kerrosten välinen vuorovaikutus tapahtuu ketjuttaen, komponenttien väliset tapahtumat kulkevat aina välittäjäpalveluiden kautta, jolloin implementointi on yksinkertaista.

Kerroksittaisen järjestelmän ja yhdenmukaisen rajapinnan rajoitteissa mainittujen itseseliteisten viestien yhteistoiminnan avulla voidaan muokata ja suodattaa kohderesurssia halutulla tavalla. Tämä on mahdollista, sillä itseseliteisten viestien semantiikka, kohderesurssi sekä haluttu toiminto ovat

näkyvissä jokaiselle välittäjäpalvelulle kaikissa sen läpikulkemissa kerroksissa.

Lisäämällä järjestelmään kerroksia kasvatetaan myös tiedonkäsittelystä johtuvaa viivettä. Tämä näkyy suoraan käyttäjälle suorituskyvyn laskuna. Välimuistin käyttöä tukevassa verkkopohjaisessa järjestelmässä tätä voidaan kompensoida hyödyntämällä välittäjäpalveluiden jaettua välimuistia. Välimuistia jakavien palveluiden asettaminen organisaation verkon rajalle voi johtaa merkittäviin suorituskyparannuksiin. [Fielding, 2000, 82-84].

## 6.6 Koodinsiirron mahdollisuus

REST-malli tarjoaa asiakas-palvelin -arkkitehtuurin asiakasosapuolelle mahdollisuuden laajentaa toiminnallisuuttaan lataamalla ja suorittamalla palvelimella sijaitsevaa ohjelmakoodia. Tämä yksinkertaistaa asiakasohjelmia vähentämällä vaadittujen etukäteen toteutettavien ominaisuuksien määrää, sekä parantaa järjestelmän laajennettavuutta. Toisaalta koodinsiirto vähentää näkyvyyttä. Näkyvyyden vähentymisestä johtuen koodinsiirron mahdollisuus on ainoa vapaaehtoinen rajoite REST:ssä.

Ristiriitaiselta kuulostavan rajoitteen vapaaehtoisuuden päämääränä on ylläpitää rajoitetta sellaisissa järjestelmän osissa, joissa se on käytössä. Esimerkiksi organisaation sisäverkossa ajettavassa asiakasohjelmassa hyödynnetään organisaation omilta palvelimilta ladattua koodia, mutta palomuurirajoitteiden vuoksi tätä toiminnollisuutta ei voida ladata sisäverkon ulkopuolisista lähteistä. Tällöin ulkopuolisesta verkosta katsottuna vaikuttaa siltä, että sisäverkossa ajettava asiakasohjelma ei tue koodinsiirtoa. Vapaaehtoisen rajoitteen ansiosta järjestelmäarkkitehtuuri voi tukea haluttua toiminnallisuutta yleisessä käyttötapauksessa, ollen kuitenkin tietoinen siitä, että joissain tapauksissa toiminnallisuus voi olla rajattua. [Fielding, 2000, 84-85].

## 7 RESTful www-sovelluspalvelurajapinnat

WWW-sovelluspalvelun toteutus tapahtuu käytännössä usein jonkin webkehitykseen soveltuvan *sovelluskehiksen* (engl. software framework) avulla. Sovelluskehiksen hallinnoidessa esimerkiksi HTTP-viestien reittejä ja tietokantayhteyttä kehittäjän vastuulle jää REST:lle ominaisten suunnitteluvälineiden lisäksi esimerkiksi sovelluksen tietoturva-arkkitehtuuri, testaus ja dokumentointi. Tässä luvussa tarkastellaan REST-kohtaisia suunnitteluparadigmoja www-sovelluspalveluiden kannalta laajentaen ja tarkentaen REST-arkkitehtuurissa määriteltyjä rajoitteita.

On huomionarvoista, että vaikka rajoitteissa puhutaan paljon asiakas-palvelin -arkkitehtuurin dynamiikasta, niin RESTful:n *www*-sovelluspalvelun toteutuksessa ei tarvitse suoranaisesti puuttua asiakasohjelman eli käyttölii-tymän toimintaan. Vaikka ohjelmistokokonaisuuden tarkoituksenmukainen toiminta riippuu myös asiakasohjelman toteutustavasta, REST:n asettamia rajoitteita ja niiden toteutustapoja hallinnoidaan pääosin palvelinohjelmis-tosta käsin. Tärkeää tämä on siksi, että *www*-sovelluspalvelulla on usein usei-ta, palvelinohjelmistoa hallinnoivasta osapuolesta riippumattomia käyttäjiä eli asiakasohjelmia.

Esimerkkikoodissa 1 kuvatut metodit toteuttavat CRUD-joukkoon (*crea-te*, *read*, *update*, *delete*) kuuluvia toimintoja. Metodeilla pyritään siis luo-maan, hakemaan, muokkaamaan ja poistamaan resursseja. Jos kuvattu ra-japinta toteutettaisiin vapaavalinnaisella ohjelmointikielellä, voitaisiin sitä käyttää kuin perinteistä ohjelmistorajapintaa. Koska kuitenkin halutaan to-teuttaa web-pohjainen *www*-sovelluspalvelurajapinta, täytyy toteutettuja me-todeja pystyä kutsumaan verkon välityksellä. Tämä onnistuu luomalla raja-pintaan resursseille URI:lla osoitettavat päätepisteet sekä määrittelemällä kyseisille resursseille mahdolliset HTTP-metodit.

## 7.1 URI

REST *www*-sovelluspalvelun URI-päätepisteiden tulee olla intuitiivisia, ku-vaavia ja helposti ymmärrettäviä. Niiden ei ole tarkoitus toimia koneellises-sa vuorovaikutuksessa, vaan mahdollistaa asiakasohjelmaa kehittävän ihmi-sen määritellä rajapintayhteyteen vaadittavia *URI-reittejä* eli URI-osoitteita [Rodriguez, 2008].

`http://www.linjaoy.fi/affB6k9/12?topic=technology`

Esimerkkikoodi 8: Huono URI.

Esimerkiksi syystä tai toisesta koneellisesti generoitujen resurssipolkujen (esimerkkikoodi 8) sekä tarpeettomien URI-kyselyosien (kohta 4.3) käyttö ei ole suositeltua, sillä ne heikentävät luettavuutta. Parhaassa tapauksessa rajapinnan URI-kokoelma dokumentoi itse itseään tarviten hyvin vähän to-dellista dokumentaatiota tai lisämäärittelyjä tuekseen.

Eräs yleistynyt tapa luoda käytettävyydeltään edistyneitä URI-reittejä on käyttää hakemistorakenteen kaltaista URI:a. Tällainen URI, kuten hyvä hakemistorakennekin, on hierarkkinen tietopuu, joka alkaa yhdestä juuresta haarautuen useaksi eri alireitiksi. Tarkastellaan esimerkkikoodissa 9 määri-

teltyä aiemmin kuvatun Linja Oy -rajapinnan URI-muotoista esitystä.

```
http://www.linjaoy.fi/trips/{tripId}
http://www.linjaoy.fi/trips/available
http://www.linjaoy.fi/trips/available/{startDate}/{endDate}
http://www.linjaoy.fi/reservation/
http://www.linjaoy.fi/reservation/{reservationId}
```

Esimerkkikoodi 9: Linja Oy -rajapinnan URI-päätepiisteet.

Esimerkissä voidaan huomata resurssien yksilöllistäminen eräänlaisen puurakenteen kautta, jonka juurina toimivat matkat ja varaukset. Aaltosulkeilla merkittyjen muuttujien avulla yksilöidään resursseja esimerkiksi tietokannassa sijaitsevan id:n tai päivämäärien mukaan. Koska resurssi voi olla resursseista koostuva kokoelma, myös esimerkiksi vapaiden matkojen kokoelma on resurssi.

URI-muotoisesta rajapinnasta puuttuu muutamia esimerkkikoodissa 1 esiintyviä asioita, kuten metodikutsujen CRUD-etuliitteet ja parametreina annettavat *reservationInfo*-muuttujat. Näistä johtuen pääasiallisia API-päätepiisteitä on myös vähemmän. Syy päätepiisteiden erilaisuuteen on se, että RESTful -rajapinnan URI-päätepiisteiden kuuluu olla ensisijaisesti resursseja, ei operaatioita [Fielding, 2000, 109-111]. Tästä syystä rajapintaa ei käännetä suoraan esimerkkikoodissa 10 kuvattuun muotoon.

```
http://www.linjaoy.fi/getAvailableTrips
```

Esimerkkikoodi 10: Operaatiokeskeinen URI.

Rodriguez [2008] antaa resurssikeskeisyyden ja hakemistorakennepohjaisuuden lisäksi muutamia lisäohjeita RESTful URI-päätepiisteiden luomiseen:

- Piilota palvelimella ajettavien skriptikielten tiedostopäätteet (.php, .js), jotta voit vaihtaa kieltä myöhemmin muuttamatta URI:a.
- Käytä vain pieniä kirjaimia.
- Käytä välilyöntien tilalla joko väliviivaa (-), tai alaviivaa (\_).
- Vältä kyselyosien (kuvattu kohdassa 4.3) käyttöä.
- Jos käytetty URI on osittaisella reitillä, palauta 404-koodin sijaan jokin oletussivu tai -resurssi.
- Substantiivien käyttö resurssipoluissa parantaa HTTP-metodien semanttista ymmärrettävyyttä.



## 7.2 HTTP

HTTP on ensisijainen RESTful-www-sovelluspalvelurajapintojen tiedonsiirtoon käytetty tiedonsiirtoprotokolla. Se noudattaa REST:n arkkitehtuaalisia rajoitteita (luku 6), kuten resurssien esitysmuotoista manipulointia ja itseselitteisiä viestejä. [Fielding, 2000, 116-121].

Koska HTTP-metodit ovat käytännössä vain semantiikkaa kuvaavia merkijonoja, HTTP-metodien (kohta 4.2) oikeellinen ja täsmällinen käyttö on avainasemassa RESTful-www-sovellusrajapinnan suunnittelussa. Rodriguez [2008] esittelee ongelman, jossa tiedonhakuun käytettävää GET-metodia käytetään tiedon luontiin (esimerkkikoodi 11).

```
GET /adduser?name=Matti HTTP/1.1
```

Esimerkkikoodi 11: Virheellinen HTTP-pyyntö aloitusrivillä, GET-metodin käyttö tiedon luonnissa.

Esimerkkikoodin 11 pyyntö korjataan muuttamalla GET-metodi semanttisesti oikeelliseksi POST-metodiksi (oletetaan, että käyttäjä lisää idempotenttisesti) ja antamalla käyttäjätiedot pyynnön viestirungossa (esimerkkikoodi 12).

```
POST /users HTTP/1.1
host: localhost
content-type: application/json
cache-control: no-cache

{"name": "Matti"}
```

Esimerkkikoodi 12: Korjattu HTTP-pyyntö.

Semanttisen virheen lisäksi GET-metodin käytöllä voi olla muitakin seuraamuksia. Esimerkiksi internetin indeksointiin käytetyt *web-indeksoijat* (engl. web crawler) saattavat linkkiä indeksoidessaan laukaista tietoa muokkavan operaation, sillä GET-metodia voidaan käyttää kirjoittamalla URI-polku suoraan www-selaimen URL-palkkiin.

Virheellisen HTTP-metodin käytöllä voi olla muitakin tietoturvakriittisiä seuraamuksia kuin tiedon eheyden vaarantaminen. Esimerkiksi kirjautumistietojen antaminen GET-parametreina URI:ssa tallentaisi jokaisella kirjautumiskerralla käyttäjän käyttäjätunnuksen sekä salasanan selaimen sivuhistoriaan.

```
GET /trips/{tripId}
GET /trips/available
GET /trips/available/{startDate}/{endDate}
GET /reservation/{reservationId}
POST /reservation/
PATCH /reservation/{reservationId}
DELETE /reservation/{reservationId}
```

Esimerkkikoodi 13: Linja Oy -rajapinnan HTTP-metodit ja kohderesurssit.

```
POST /reservation/ HTTP/1.1
host: localhost
content-type: application/json
cache-control: no-cache

{
  "info": {
    "name": "Meikäläinen Matti",
    "seats": 2
  }
}
```

Esimerkkikoodi 14: Matkan varaus Linja Oy:n palvelusta.

```
HTTP/1.1 400
status: 400
content-encoding: gzip
content-type: application/json; charset=utf-8
cache-control: max-age=16400
connection: keep-alive

{
  "messages": [
    {
      "message": "reservation was succesfully created",
      "type": "resource_created",
      "resource": "reservation/12"
    }
  ]
}
```

Esimerkkikoodi 15: HATEOAS:n toteutus.

Kohdassa 7.1 huomattiin, että URI-päätepisteistä puuttuu CRUD:n mukaiset funktioetuliitteet. Nämä määritellään HTTP-metodeilla, joista suurin osa kuuluu CRUD-joukkoon. Puuttuvat reservationInfo -muuttujat annetaan POST- ja PATCH-metodeilla tehtyjen pyyntöjen viestirungossa (esimerkkikoodi 14).

Kohdassa 6.4 nostettiin esiin hypermedia sovelluksen ajurina -rajoitteen vähäinen käyttö. Linja Oy:n tapauksessa rajoitteen toteuttaminen onnistuisi esimerkiksi palauttamalla luodun tilauksen yhteydessä HTTP-vastauksen viestirungossa kyseisen tilauksen resurssipolun eli URI:n, joten lisäämme kyseisen toiminnollisuuden esimerkkikoodissa 13 kuvattuun rajapintaan. Toiminnollisuutta havainnollistetaan esimerkkikoodissa 15.

### 7.3 HTTP-vastaukset

HTTP-vastaus sisältää aina statuskoodin [Fielding and Reschke, 2014b], esimerkiksi useimmille tutun 404 (Not Found) -koodin. Statuskoodin lisäksi HTTP-vastaus sisältää tyypillisesti metatietoja (esim. mediatyyppi) vastauksen sisällöstä sekä viestirungon sisällön.

Fielding ja Reschke [2014b] määrittelevät HTTP:n statuskoodien jakautuvan viiteen eri kategoriaan: informatiivisiin (1xx), onnistuneisiin (2xx), uudelleenohjattaviin (3xx), asiakasvirheisiin (4xx) ja palvelinvirheisiin (5xx).

Jokaiselle statuskoodille ei kannata pyrkiä etsimään käyttötarkoitusta, vaan parasta olisi pitää www-sovelluspalvelurajapinnan palauttamien statuskoodien joukko suhteellisen suppeana. Statuskoodien liiallinen laajuus johtaa siihen, että asiakasohjelman kehittäjä joutuu ottamaan tai pahemmassa tapauksessa ei huomaa ottaa kehitystyössään huomioon suurta määrää erilaisilla statuksilla kuvattuja tilanteita, esimerkiksi käyttöliittymässä näytettäviä virheilmoituksia varten.

Statuskoodi itsessään ei aina riitä kuvaamaan pyynnön onnistumista, vaan lisäksi saatetaan tarvita vapaamuotoista lisäselvitystä esimerkiksi ilmenneestä virheestä. Tämä annetaan vastauksen viestirungossa. Tällaisessa viestissä voidaan esimerkiksi ilmoittaa, miksi vastattiin kyseisellä statuskoodilla tai mitä jonkin virheen korjaamiseksi voitaisiin tehdä (esimerkkikoodi 16). HTTP-vastauksissa tulevien viestirunkojen tulee olla johdonmukaisia läpi järjestelmän. Tämä onnistuu parhaiten niin, että päätetään järjestelmälle yhteinen joukko mahdollisia statuskoodeja. Tämän lisäksi määritellään jokin yleinen, viestirungossa käytettävä, mediatyyppinen tietomuoto tai -rakenne, kuten JSON tai XML.

```
HTTP/1.1 400
status: 400
content-encoding: gzip
content-type: application/json; charset=utf-8
cache-control: no-cache
date: Mon, 04 Dec 2017 12:48:27 GMT
connection: keep-alive

{
  "messages": [
    {
      "message": "name field in request body required",
      "type": "empty_body_field"
    },
    {
      "message": "message string length must be between 1 and
        255",
      "type": "invalid_message_length"
    }
  ]
}
```

Esimerkkikoodi 16: Esimerkinomainen HTTP-viestirunkorakenne.

Erityisen tarkkana kannattaa olla asiakasohjelmasta johtuvien virheiden kanssa, sillä niiden havainnointi ja selkeä dokumentointi voi mahdollisesti nopeuttaa ja suoraviivaistaa asiakasohjelmaa kehittävän sovelluskehittäjän työprosessia huomattavasti.

## 8 Yhteenveto

Tutkielmassa tarkasteltiin REST-sovellusarkkitehtuurimallia ja sen liitännäisteknologioita www-sovelluspalveluiden näkökulmasta. Tämän lisäksi havainnollistettiin käsiteltyjä aiheita kuvitteelliselle Linja Oy:lle luodun www-sovelluspalvelurajapinnan avulla. Tutkielman aikana aiheita käsiteltiin progressiivisesti esimerkkien avulla alkaen www-sovelluspalveluiden perusteista, päättyen REST-arkkitehtuurimallin määrittelemiin rajoitteisiin ja REST:n käyttöön www-sovelluspalvelurajapinnan suunnittelussa.

WWW-sovelluspalveluita ja -sovelluspalvelurajapintoja käsiteltiin tutkielman alussa melko popularisoidusti. Tällä pyrittiin luomaan lukijalle vakaa

ymmärrys kyseisten teknologioiden käyttötarkoituksista. Käsittelemällä HTTP-protokollan perusteita yleisellä tasolla pohjustettiin erityisesti oikeellisten HTTP-metodien käyttöä RESTful-www-sovelluspalvelurajapinnoissa.

REST-arkkitehtuurimalliin kuuluvia rajoitteita käsitellessä huomattiin rajoitteiden keskittyvän tiettyjen päämäärien ympärille. Suurin osa rajoitteista pyrkii edistämään järjestelmän suorituskykyä, skaalautuvuutta tai kehityskelpoisuutta esimerkiksi löyhien sidonnaisuuksien avulla. Useat rajoitteista ovat kuitenkin pohjimmiltaan kompromisseja ja saattavat näin esimerkiksi parantaa sidonnaisuutta suorituskyvyn kustannuksella.

Rajoitteita käsiteltäessä havaittiin myös, että REST:n keskittyessä yhdenmukaisen rajapinnan toimintaan ja rajoitteisiin ottaa se kantaa rajapinta-arkkitehtuurin lisäksi järjestelmän kokonaisarkkitehtuuriin. Vaikka REST määrittelee ohjelmistoarkkitehtuurin rakennetta ja ominaisuuksia osittain melko tarkasti, se ei ota kantaa mallin mukaisesti luodun järjestelmän toteutustekniikoihin tai teknisiin yksityiskohtiin, jolloin RESTful-rajapinnan toteutuksessa käytetyt teknologiat voidaan valita tarkoituksenmukaisesti. Tällöin, yhdistettynä HTTP:n käyttöön tiedonsiirtoprotokollana, REST toteuttaa osaltaan www-sovelluspalveluille ominaisen alustariippumattomuuden.

Tutkielman rajatusta laajuudesta johtuen tutkielmassa ei käsitelty tarkemmin muita www-sovelluspalvelurajapintojen toteutuksessa oleellisia aiheita, kuten autentikaatiota, ohjelmistotestausmetodeja (esimerkiksi automatisoitu yksikkötestaus), CSRF/XSS -haavoittuvuutta tai HTTP:n *kuljetuskerroksen turvallisuus* -protokollalla (engl. transport layer security, tai TLS) salattua versiota: HTTPS:ää. Samasta syystä ei myöskään käsitelty vaihtoehtoisia teknologioita, kuten RCP:tä tai SOAP:ia.

REST-arkkitehtuurimallissa määriteltyjen rajoitteiden sekä tutkielmassa käsiteltyjen www-sovelluspalvelurajapintakohtaisten suunnitteluperiaatteiden seuraaminen ei yksin riitä hyvän www-sovelluspalvelurajapinnan tai -sovelluspalvelun toteutukseen. RESTful-rajapinnan kehittäjän on aina huomioitava myös muut joustavan ja hyvän suunnittelun keinot, sekä arvioitava käytettävien toteutustekniikoiden ja -teknologioiden osuus kokonaisarkkitehtuurissa. Varsinkin tilanteessa, jossa kehitetään www-sovelluspalvelua tietoturvakriittisen tiedon julkaisemiseen rajatulle asiakaskunnalle, on otettava huomioon rajapinnan tietoturva-aspekti. Tietoturvan kokonaisvaltainen kartoitus RESTful-rajapinnoissa olisikin tärkeä jatkotutkimusaihe.

## Viitteet

Alarcón, R., and Wilde, E. 2010. Restler: crawling restful services. In: *Proc.*

- of the 19th International Conference on World Wide Web*, 1051–1052.
- Alarcon, R., Wilde, E., and Bellido, J. 2010. Hypermedia-driven restful service composition. In: *Proc. of ICSOC Workshops*, 111–120.
- Belshe, M., Peon, R., and Thomson, M. 2015. *Hypertext transfer protocol version 2 (http/2)* (RFC nro 7540). RFC Editor. Internet Requests for Comments. Checked 14.12.2017, Available <http://www.rfc-editor.org/rfc/rfc7540.txt>.
- Berners-Lee, T., Fielding, R., and Masinter, L. 2005. *Uniform resource identifier (uri): Generic syntax* (STD nro 66). RFC Editor. Internet Requests for Comments. Checked 14.12.2017, Available <http://www.rfc-editor.org/rfc/rfc3986.txt>.
- Braden, R. 1989a. *Requirements for internet hosts - application and support* (STD nro 3). RFC Editor. Internet Requests for Comments. Checked 14.12.2017, Available <http://www.rfc-editor.org/rfc/rfc1123.txt>.
- Braden, R. 1989b. *Requirements for internet hosts - communication layers* (STD nro 3). RFC Editor. Internet Requests for Comments. Checked 14.12.2017, Available <http://www.rfc-editor.org/rfc/rfc1122.txt>.
- Brown, A., and Haas, H. 2004. *Web services glossary* (W3C Note). Checked 14.12.2017, Available <http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/>.
- Dusseault, L., and Snell, J. 2010. *Patch method for http* (RFC nro 5789). RFC Editor. Internet Requests for Comments. Checked 14.12.2017, Available <http://www.rfc-editor.org/rfc/rfc5789.txt>.
- Feng, X., Shen, J., and Fan, Y. 2009. Rest: An alternative to rpc for web services architecture. In: *Proc. of the First International Conference on Future Information Networks*, 7–10.
- Fielding, R. 2000. Rest: Architectural styles and the design of network-based software architectures. *Doctoral dissertation, Department of Information and Computer Science, University of California*.
- Fielding, R. 2008. *REST APIs must be hypertext-driven*. Checked 3.12.2017, Available <http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>.

- Fielding, R., Nottingham, M., and Reschke, J. 2014. *Hypertext transfer protocol (http/1.1): Caching* (RFC nro 7234). RFC Editor. Internet Requests for Comments. Checked 14.12.2017, Available <http://www.rfc-editor.org/rfc/rfc7234.txt>.
- Fielding, R., and Reschke, J. 2014a. *Hypertext transfer protocol (http/1.1): Message syntax and routing* (RFC nro 7230). RFC Editor. Internet Requests for Comments. Checked 29.11.2017, Available <http://www.rfc-editor.org/rfc/rfc7230.txt>.
- Fielding, R., and Reschke, J. 2014b. *Hypertext transfer protocol (http/1.1): Semantics and content* (RFC nro 7231). RFC Editor. Internet Requests for Comments. Checked 4.12.2017, Available <http://www.rfc-editor.org/rfc/rfc7231.txt>.
- Freed, N., and Borenstein, N. S. 1996. *Multipurpose internet mail extensions (mime) part two: Media types* (RFC nro 2046). RFC Editor. Internet Requests for Comments. Checked 3.12.2017, Available <http://www.rfc-editor.org/rfc/rfc2046.txt>.
- Garlan, D., and Shaw, M. 1993. An introduction to software architecture. In: V. Ambriola and G. Tortora (eds.), *Advances in Software Engineering and Knowledge Engineering*. World Scientific, 1–39.
- Liu, S., Li, Y., Sun, G., Fan, B., and Deng, S. 2017. Hierarchical rnn networks for structured semantic web api model learning and extraction. In: *Proc. of 2017 IEEE International Conference on Web Services (ICWS)* 708–713.
- Medjahed, B. 2004. *Semantic web enabled composition of web services* (unpublished dissertation).
- Mumbaikar, S., and Padiya, P. 2013. Web services based on soap and rest principles. In: *International Journal of Scientific and Research Publications* 3, 5, 1–4.
- Parziale, L., Liu, W., Matthews, C., Rosselot, N., Davis, C., Forrester, J., Britt, D. T., et al. 2006. *TCP/IP Tutorial and Technical Overview*, IBM Redbooks.
- Pautasso, C. 2009. Restful web service composition with bpel for rest. In: *Data and Knowledge Engineering* 68, 9, 851–866. Elsevier.

- Pautasso, C. 2014. Restful web services: principles, patterns, emerging technologies. In: A. Bouguettaya, Q. Z. Sheng, and F. Daniel (eds.), *Web Services Foundations*. Springer, 31–51.
- Pautasso, C., Zimmermann, O., and Leymann, F. 2008. Restful web services vs. big’web services: making the right architectural decision. In: *Proc. of the 17th International Conference on World Wide Web*, 805–814.
- Rodriguez, A. 2008. *Restful web services: The basics*. Checked 14.12.2017, Available <https://www.ibm.com/developerworks/library/ws-restful/ws-restful-pdf.pdf>.
- Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., and Xu, X. 2014. Web services composition: A decade’s overview. In: *Information Sciences* 280(Supplement C), 218–238.
- Thaler, D., Hansen, T., and Hardie, T. 2015. *Guidelines and registration procedures for uri schemes* (BCP nro 35). RFC Editor. Internet Requests for Comments. Checked 14.12.2017, Available <https://www.rfc-editor.org/rfc/rfc7595.txt>.
- Vinoski, S. 2008. RESTful Web Services Development Checklist. In: *IEEE Internet Computing* 12, 6, 96–95.
- Waldo, J., Wyant, G., Wollrath, A., and Kendall, S. 1996. A note on distributed computing. In: *Proc. of International Workshop on Mobile Object Systems*, 49–64.



# Sähköinen ostolaskujärjestelmä: Ostolaskujen käsittelyprosessin automaatio

Joni Finckenberg

## Tiivistelmä.

Tämän tutkielman tarkoituksena on esitellä sähköistä ostolaskujärjestelmää, ostolaskujen käsittelyprosessia ja sen tuomia mahdollisuuksia, sekä analysoida, miten ostolaskujen käsittelyprosessi voidaan sähköistää ja automatisoida. Ostolaskujen käsittely on samaa kaavaa toistavaa manuaalista työtä, joka tarjoaa hyvän mahdollisuuden automatisoinnille. Tutkielmassa käsitellään teorian lisäksi teknistä näkökulmaa, nostamalla hankinnasta maksuun ratkaisua tarjoavaan Baswaren ratkaisuja esiin.

Tutkielman menetelmänä ovat kirjallisuuskatsaus ja Baswaren ratkaisut kuvattavana tapauksena. Tutkielmassa verrataan Baswaren ratkaisuja siihen, mitä saman aiheen kirjallisuudessa kerrotaan. Tutkielma osoittaa, että ostolaskujen käsittelyprosessin ymmärtäminen mahdollistaa tehokkaiden digitaalisten prosessien rakentamisen. Sähköisillä ja automatisoiduilla prosesseilla voidaan saavuttaa mittavia säästöjä ja tehokkuuden parannuksia. Sähköistämisen myötä prosessi on mahdollista automatisoida joko osittain tai kokonaan.

**Avainsanat ja -sanonnat:** Sähköinen ostolaskujärjestelmä, sähköinen taloushallinto, digitaalinen taloushallinto, ostolaskujen käsittelyprosessi, ostolaskuautomaatio, verkkolasku, e-invoice.

## 1. Johdanto

Sähköiset ostolaskujärjestelmät ovat nykypäivää useassa yrityksessä. Sähköisten ostolaskujärjestelmien pääasiallinen tavoite on helpottaa ja automatisoida laskujen käsittelyprosessia. Paperilaskujen vaihtaminen verkkolaskuihin tuo yleisesti ottaen suuria kustannussäästöjä. Laskujen vastaanottaminen suoraan automatisoituun järjestelmään nostaa tuottavuutta ja pienentää laskujen käsittelykustannuksia.

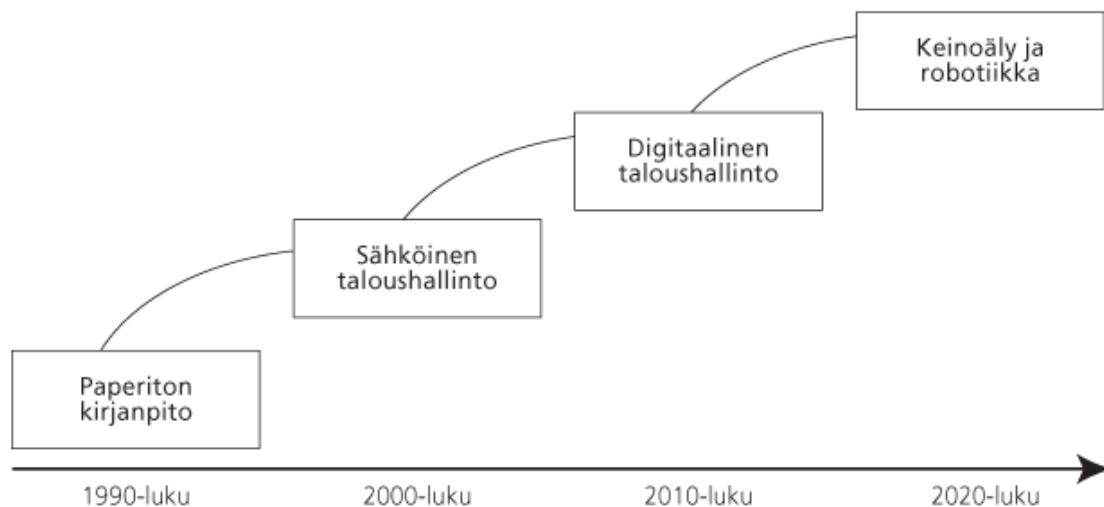
Nykypäivän kovassa yritysten välisessä kilpailussa ja taloustilanteessa yritykset pyrkivät kustannustehokkuuteen ja säästöihin. Digitaalinen taloushallinto ja ostolaskujärjestelmä ovat isossa osassa yritysten tietojärjestelmiä, joilla voidaan saavuttaa huomattavia kustannussäästöjä. Suomalainen lainsäädäntö on mahdollistanut sähköisen taloushallinnon vuodesta 1997 alkaen. Digitaalinen taloushallinto on yleistynyt Suomessa ennustettua hitaammin. Tästä johtu-

en Suomi on menettänyt ensimmäisen sijansa sähköisten taloushallintosovellusten käyttäjänä. [Lahti ja Salminen 2014.]

Tutkielmassa perehdytään sähköiseen ja digitaaliseen taloushallintoon ja erityisesti ostolaskukäsittelyjärjestelmään. Keskeisin osuus tutkielmasta pitää sisällään sähköisen ostolaskukäsittelyprosessin välivaiheet ostolaskun käsittelystä aina kirjanpitoon ja maksatukseen asti ja mahdollisuudet niiden automatisointiin. Tutkielman viimeisessä luvussa tehdään loppupäätelmät ja yhteenveto.

## 2. Digitaalinen taloushallinto

Kuvasta 1 voidaan päätellä, että digitaalisen taloushallinnon perusta on luotu 1990-luvulla. Paperiton kirjanpito tarkoittaa lakisääteisten tositteiden säilyttämistä sähköisessä muodossa eikä varsinaisesti vielä digitaalista taloushallintoa. Suurin osa tiedosta on saatu 2000-luvun alkupuolella kulkemaan sähköisessä muodossa. Tulevaisuudessa oletetaan keinoälyn ja robotiikan kehittävän automaatiota uudelle tasolle. [Lahti ja Salminen 2014.] Baswaren ratkaisuihin keinoälystä on jo merkkejä. Baswaren smart coding perustuu algoritmiin, joka tarjoaa käyttäjälle ehdotuksen aikaisempien mallipohjien, automaattiosäätöjen ja historian perusteella [Basware.com 2017].



**Kuva 1.** Sähköisen taloushallinnon kehitys Suomessa [Lahti ja Salminen 2014].

Yrityksen taloushallintoon kuuluu monia tärkeitä tehtäviä, joista yksi on ostolaskujen käsittely. Taloushallintojärjestelmällä organisaatiot voivat seurata taloudellisia tapahtumia ja tehdä vaadittuja raportteja eri sidosryhmille. Digitaalinen taloushallinto kattaa kaikki taloushallinnolle tyypilliset tehtävät, ja sen tärkeimpänä tehtävänä on kaikkien tietovirtojen ja käsittelyvaiheiden automatisointi ja käsittely digitalisessa muodossa ilman paperia. Käytännössä

taloushallinto määritellään järjestelmäksi, joka sisältää toisiinsa liittyviä komponentteja, jotka on integroitu toimimaan yhdessä saavuttaakseen niille asetetut tavoitteet. [Lahti ja Salminen 2014.] Digitaalisen taloushallinnon tärkein toimintaperiaate on, että järjestelmään syötetyt tiedot ovat yrityksen ja muiden sidosryhmien käytettävissä [Yrittajat.fi 2014].

Digitaalinen taloushallinto tarjoaa pohjan laskujen sähköiseen ja automatisoituun käsittelyyn. Automatisoinnin tavoitteena on poistaa turhat ja päällekkäiset käsittelyvaiheet sähköisessä muodossa olevassa tiedonkäsittelyssä. Digitaalisten taloushallinto- ja ostolaskujärjestelmien myötä taloushenkilöiden ja ostolaskun käsittelijöiden työt voivat tehostua huomattavasti. Automaatio vähentää virheiden riskiä ja sähköisesti hoidettavat tehtävät ovat ympäristöystävällisempiä. [Lahti ja Salminen 2014.]

## **2.1. Sähköinen tiedonsiirto**

Digitaalinen taloushallinto ei ole mahdollista ilman tiedonsiirtoa yritysten välillä. Sähköinen tiedonsiirto on ollut mahdollista yli 30 vuoden ajan. Organisaatioiden väliselle tiedonsiirrolle on olemassa *organisaatioiden välinen tiedonsiirto (OVT)* (eng. electronic data interchange, EDI), joka tarjoaa perustan sähköiselle mahdollisimman pitkälle automatisoidulle tiedonvaihdolle yritysten välillä. [Lahti ja Salminen 2014.] UN/EDIFACT (The United Nations EDI for Administration, Commerce, and Transportation) määrittelee OVT:n standardiformaatiksi yritysten välisessä tiedonsiirrossa. Standardilla pyritään siihen, että tiedonsiirto vaatii mahdollisimman vähän manuaalista puuttumista. Yleisellä tasolla tarkastellen OVT toimii siten, että yrityksen tietojärjestelmät vaihtavat ennalta määritellyjä viestejä keskenään. [Ahadi 2004.]

OVT:n käytön hyödyt ovat samankaltaisia kuin internetin tuomat hyödyt esimerkiksi pikaviestinnässä: tietojenkäsittely nopeutuu, virheet vähenevät, kustannukset pienenevät, tiedonhallinta ja arkistointi paranevat [Lahti ja Salminen 2014]. Näiden lisäksi hyötyinä nähdään tietoturvallisuuden parantuminen, kontrolli, ja yritysten sisäisen tiedonvaihdon parantuminen [Ahadi 2004]. Teknisten hyötyjen lisäksi OVT tehostaa liiketoimintaan liittyviä hyötyjä, kuten kilpailukykyä ja liiketoimintaprosessien tehostumista. Yrityksissä OVT:ta pidetään taloudellisesti ja liiketoiminnan kannalta merkittävänä työkaluna. Suurimpina esteinä OVT:n käyttöön otolle nähdään teknologian monimutkaisuus ja huonot yhteensopivuudet nykyisten tietojärjestelmien kanssa. Yritykset käyttävät useasti ulkopuolisia operaattoreita tiedon siirtämiseen ja muuntamiseen. [Lahti ja Salminen 2014.]

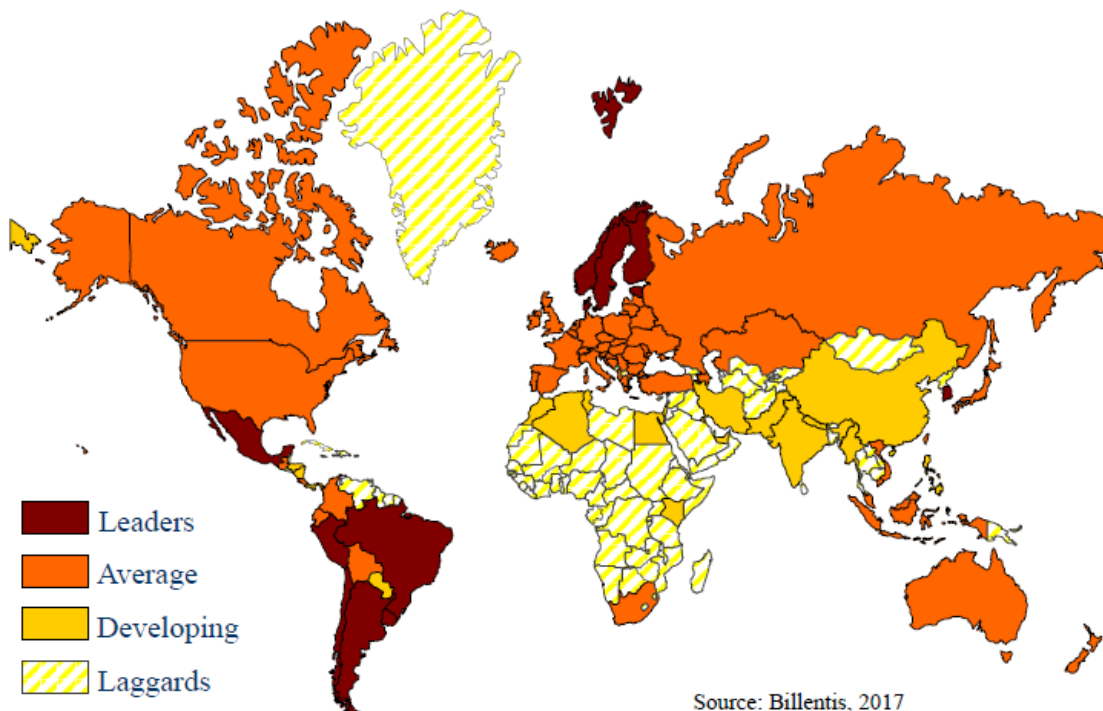
## 2.2. Laskut sähköisessä muodossa

Sähköinen ostolaskujen käsittely vaatii sähköisessä muodossa olevia laskuja. Verkkolasku on sähköisessä muodossa lähetettävä ja vastaanotettava sanoma, joka sisältää vähintään kaikki samat tiedot kuin perinteinen paperilasku [Lahti ja Salminen 2014]. Verkkolaskutus on keskeinen osa sähköistä ostolaskujärjestelmää, sillä yritykset pyrkivät automatisoimaan laskujen käsittelyä sähköisten laskujen avulla [Vesela and Radimersky 2014]. Vielä viime vuosikymmenellä maailmassa oli alle 15 verkkolaskujen toimittajaa. Toimittajien määrän on enustettu olevan lähivuosina yli 1500. Verkkolaskutusmarkkinoiden arvioksi on laskettu vuonna 2017 yli 3,3 miljardia euroa. Pelkästään paperilaskujen vaihtaminen verkkolaskuihin mahdollistaa yrityksille 60-80% kustannussäästöt. [Koch 2017.] Laskelmien mukaan sähköinen lasku voi vähentää jopa 75% kuluja yhtä laskua kohden [Vesela and Radimersky 2014]. Sähköisten laskujen aikaisemmat tutkimukset ovat osoittaneet samansuuruisia jopa 70% kustannussäästöjä [Netter *et al.* 2010].

Finanssialan keskusliitto on selvittänyt verkkolaskutuksen liittyviä ympäristöhyötyjä. Paperisen laskun osalta hiilijalanjälki on noin 500 ja 2000 hiilidioksidigramman välillä ja verkkolaskulla hiilijalanjälki on tehokkaimmillaan vain noin 150 grammaa. Suurimmat päästösäästöt saavutetaan, kun yritys siirtyy verkkolaskujen vastaanottajaksi, jolloin manuaaliset paperilaskuun liittyvät työtehtävät poistuvat ja päästöt vähenevät jopa noin neljäsosaan paperilaskuun verrattuna. [Hellgren ja Tenhunen 2010.]

Kilpailu verkkolaskutus- ja ostolaskujärjestelmämarkkinoilla on tiukentunut, mutta silti arvioiden mukaan yli 90% laskuista koko maailmassa käsitellään edelleen täysin manuaalisesti [Koch 2017]. Tämä tarjoaa huomattavat mahdollisuudet laskuautomaation markkinoiden laajentamiselle. Suomessa tilanne on kuitenkin täysin erilainen. Tilastokeskuksen vuonna 2013 teettämässä tutkimuksessa 60% yrityksistä käyttää verkkolaskuja sähköisen laskun vastaanotossa [Tilastokeskus 2013].

Kuvasta 2 huomataan, kuinka vastaanottavia eri maat ja maanosat ovat verkkolaskutukselle. Kuvasta voidaan päätellä, että Pohjoismaat ja Etelä-Amerikka ovat johtavia maita verkkolaskutuksessa. Afrikassa ja Aasiassa on vielä paljon kehitettävää.



**Kuva 2.** Markkinoiden kypsyys verkkolaskuille eri maissa ja maanosissa [Koch 2017].

Euroopan unioni on omalla toiminnallaan mahdollistanut yrityksille mahdollisuuden valita monista valtuutetuista verkkolaskutoimittajista [Lee and Wang 2013]. Euroopan unionin tavoitteena digitaalisessa agendassa on vuoteen 2020 mennessä kaikkien valtioiden laskut liikkuisivat pelkästään sähköisesti [Vesela and Radimersky 2014]. Suomessa tärkein verkkolaskutusta edistävä toimija on Tietoyhteiskunnan kehittämiskeskus ry (TIEKE). Yhdistyksen ja muiden toimijoiden ansiosta Suomen verkkolaskutus on todella hyvällä tasolla. Suomen valtio on osallistunut verkkolaskutustalkoisiin omalla panoksellaan kehottamalla toimittajiaan lähettämään verkkolaskuja vuodesta 2008 alkaen. Monet suuret organisaatiot ja yritykset ovat päättäneet vastaanottaa vain verkkolaskuja. [Lahti ja Salminen 2014.]

Verkkolaskutuksessa toimittajan lähettämät laskun tiedot ja kuvat viedään suoraan ostolaskujärjestelmään. Verkkolaskutus mahdollistaa laskun automaattisen tiliöinnin, täsmäytyksen tilaukseen tai sopimukseen. Laskun kierrätyks-, tarkastus- ja hyväksyntäprosessi voidaan automatisoida. Järjestelmän toimesta voidaan automatisoida toimenpiteitä, mitkä ennen vaativat henkilöresursseja. Tämä luo perustan taloushallinnon kustannustehokkuuden kasvatamiselle. Henkilöresurssien työnkuva muuttuu manuaalisten transaktioiden suorittamisesta enemmänkin poikkeustapausten käsittelyyn. [Lahti ja Salminen 2014.]

### 3. Sähköinen ostolaskujärjestelmä

Ostolaskujen käsittely on hyvin perinteinen ja vanha taloushallinnon työtehtävä. Ostolaskuprosessi alkaa hankintaehdotuksen tai tuotteen ostamisesta. Prosessi pitää sisällään tapahtumat ostosta aina maksuun asti. Ilman sähköistä ostolaskujen käsittelyä laskujen käsittelyn automaation rakentaminen ei ole mahdollista. Suurimmat hyödyt saavutetaan vain, jos ostolaskujen käsittelyprosessi sähköistetään. [Lahti ja Salminen 2014.] Sähköinen laskujen käsittely on ollut mahdollista yli 20 vuotta [Koch 2017]. Suomalaiset ohjelmistojen kehittäjät yritykset ovat tarjonneet pitkään sähköistä taloushallinto- ja ostolaskujen käsittelyjärjestelmää. Baswarea pidetään alansa ainoana suomalaisena kansainvälisesti menestyneenä yrityksenä. [Lahti ja Salminen 2014.]

Lukuisissa ERP-järjestelmissä (Enterprise Resource Planning) on sisäänrakennettu taloushallinto ja ostolaskujen käsittelyjärjestelmä. ERP-järjestelmä yhdistää kaikki yrityksen toiminnot yhteen tietovarastoon. Tietoja käytetään hyödyksi erityisesti yrityksen päätöksenteossa. Tiedonjakaminen eri yksiköiden ja toimitusketjun jäsenten välillä antaa monia etuja tiedonhallinnassa aina logistiikkatietojen integroimisesta taloudelliseen tuottavuuteen ja tuotekehitykseen. [Suprpto *et al.* 2017.] Monet yritykset käyttävät tästä huolimatta erillistä järjestelmää ostolaskujen käsittelyyn. Nykypäivän trendi on ollut, että yritykset siirtyvät käyttämään ERP:stä löytyvää järjestelmää, jotka ovat kehittyneet toiminnallisuudeltaan vastaamaan paremmin erillistä ostolaskujen käsittelyjärjestelmää. ERP:n oman ostolaskujen käsittelyjärjestelmän käyttöönotto on yleensä helpompaa, koska integraatioita ei tarvitse rakentaa ulkoiseen järjestelmään ja näin ollen pystytään suoraan käyttämään hyödyksi siellä ylläpidettyjä tietoja. [Lahti ja Salminen 2014.]

Ostolaskukäsittelyprosessi paperilaskuilla on aikaa vievää ja täysin manuaalinen työtehtävä. Paperisten dokumenttien käsittelyyn sisältyy paljon riskejä, kuten dokumenttien katoaminen. Digitaalisen tiedon siirtäminen on tehokkaampaa ja nopeampaa verrattuna fyysisessä muodossa olevan tiedon siirtämiseen. Samalla erilaisten resurssien ja arkistointitilan tarve vähenee huomattavasti. [Lahti ja Salminen 2014.]

### 4. Erilaiset ostolaskuprosessit

Ostolaskun käsittelyssä on oleellista, että ostetuista palveluista tai tuotteista tuotetaan lasku, joka kuvaa, mitä tulee maksaa, mukaan lukien erilaiset säännöt ja ehdot. Laskun luominen voi olla monimutkainen prosessi, sillä eri järjestelmät luovat laskuja eri tavalla, mutta kaikilla laskuilla on kuitenkin yhteisiä piirteitä. [Fernandez and Yuan 2009.]

Taloushallinnon tehtävistä kaikista eniten resursseja kuluttavana työtehtävänä pidetään ostolaskujen käsittelyä. Ostolaskujärjestelmiä käyttävät yritykset pyrkivät verkkolaskutukseen, jotta automatisointi olisi mahdollista. [Lahti ja Salminen 2014.] Tämä ei kuitenkaan kaikkien yritysten kohdalla ole mahdollista, sillä monissa yrityksissä erityisesti kuvassa 2 keltaisella merkityissä maissa olevat toimittajat lähettävät pääsääntöisesti vain paperimuodossa olevia laskuja [Koch 2017]. Yrityksillä on käytössä skannauspalveluita, jossa paperilaskut lähetetään tiettyyn osoitteeseen skannattavaksi sähköiseen muotoon. Skannauksen aikana pyritään poimimaan laskusta oleelliset tiedot mukaan. Skannaus on manuaalista työtä, eikä se tarjoa niin hyviä mahdollisuuksia laskuautomaation rakentamiselle. [Lahti ja Salminen 2014.]

Yleinen ostoprosessi jaetaan useampaan eri vaiheeseen [Lahti ja Salminen 2014]:

1. Tilaus- ja toimitusprosessi.
2. Ostolaskun vastaanotto.
3. Ostolaskun tiliöinti.
4. Ostolaskun täsmäytys tilaukseen/sopimukseen tai ostolaskun tarkistus ja hyväksyntä.
5. Maksatus.
6. Täsmäytykset ja jaksotukset oikeille tilikausille.
7. Arkistointi.

Yllä olevat vaiheet kuvaavat taloushallinnon tehtäviä. Prosessi lähtee liikkeelle, kun ostolasku saapuu, ja päättyy, kun ostolasku on maksettu toimittajalle, kirjattu kirjanpitoon ja arkistoitu. Hankinnan näkökulmasta prosessi alkaa huomattavasti aikaisemmin. Hankinta lähtee liikkeelle sopivan toimittajan löytämisestä ja tarjouspyynnön tekemisestä. Hyväksytyn tarjouksen perusteella tehdään sopimus toimittajan kanssa. Tilaukseen tai sopimukseen liittyvä laskukäsittely voi olla vastaanottoon perustuva, jolloin laskun täsmäytys voidaan automaattisesti kohdentaa vastaanotettuun määrään. [Lahti ja Salminen 2014.] Toimittajan valintaan liittyy monimutkainen prosessi, johon on saatavilla digitaalisia ratkaisuja. Työkalut tarjoavat ratkaisuja tarjouskilpailun, neuvottelun, toimituksien ja tilauksien hallintaan. Sovellukset mahdollistavat myös nopean vastauksien ja palautteen vaihdon toimittajien välillä. [Lee and Wang 2013.].

### **3.1 Manuaalinen ostolaskuprosessi**

Aikaisemmin ostolaskuprosessi on tapahtunut ilman sähköisiä ja digitaalisia järjestelmiä. Perinteinen manuaalinen ostolaskuprosessi on edennyt seuraavalla tavalla [Lahti ja Salminen 2014]:

1. Ostolasku saapuu paperilla.

2. Lasku toimitetaan asiataarkastajalle.
3. Asiatarkastaja merkkää laskun tarkastetuksi.
4. Asiatarkastaja toimittaa laskun hyväksyjälle.
5. Hyväksyjä merkkää laskun hyväksytyksi.
6. Hyväksyjä toimittaa laskun ostoreskontranhoidajalle.
7. Ostoreskontranhoidaja tallentaa manuaalisesti laskun perustiedot ja tiliöinnin ostoreskontraan.
8. Ostoreskontranhoidaja arkistoi paperilaskun.
9. Ostolaskuista muodostetaan maksuaineisto, joka siirretään maksettavaksi pankkiin.

Tästä on selkeästi huomattavissa, kuinka paljon resursseja ja työtä vaativa ostolaskujen käsittelyprosessi on. Suurimpia ongelmia ovat hidas laskujen kierto, laskujen näkyvyys kirjanpidossa laskujen kirjauksen jälkeen, monivaiheiset ja manuaaliset toimenpiteet [Lahti ja Salminen 2014].

Ostolaskujen käsittelyprosessi sähköisessä muodossa tapahtuu samantyyppisesti kuin perinteinen ostolaskuprosessi. Neljän silmän eli kahden käsittelijän periaate liittyy oleellisesti molempiin prosesseihin. Kahden käsittelijän periaatteella tarkoitetaan, että laskulla on kierron aikana tarkastus sekä hyväksyntä. [Ahokas 2012.] Manuaalisen laskujen käsittelyprosessin ollessa hidas ja resursseja vaativa työtehtävä, sähköinen laskujen käsittely on parhaimmillaan mahdollisimman automaattinen, nopeampi ja paremmassa kontrollissa [Lahti ja Salminen 2014].

### **3.2 Sähköinen ostolaskuprosessi ilman tilausta tai sopimusta**

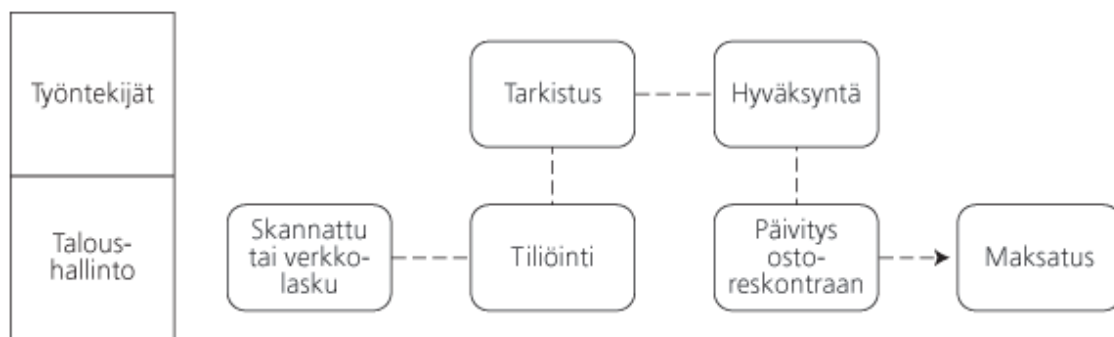
Sähköisellä ostolaskuprosessilla ilman tilausta tai sopimusta tarkoitetaan, että ostolaskua ei kohdisteta sähköisesti saatavilla olevaan tilaukseen tai sopimukseen. Sähköinen ostolaskuprosessi on jaettu vähempään määrään eri vaiheita kuin perinteinen ostolaskuprosessi [Lahti ja Salminen 2014]:

1. Ostolasku vastaanotetaan ostolaskujärjestelmään. Lasku voi olla verkkolasku tai skannattu sähköiseen muotoon.
2. Ostolaskulle suoritetaan tiliöinti automaattisesti, jos se on mahdollista.
3. Ostolasku lähtee kiertoon tarkastukseen ja hyväksyntään. Kiertoon lähetys voi olla manuaalinen tai automaattinen.
4. Ostolaskuille suoritetaan tarkastus ja hyväksyntä.
5. Hyväksytyt laskut kirjataan sähköisesti ostoreskontraan.
6. Ostoreskontran kautta suoretaan laskun maksatus.

Basware määrittelee standardin ostolaskuprosessin samantylaisiin vaiheisiin, kuten edellä ja kuvassa 3 on kuvattu. Baswaren tarjoama Purchase-to-Pay, jatkossa Basware P2P, -sovelluksen standardi ostolaskuprosessi pitää sisällään



ostolaskujen vastaanoton, validoinnin, hyväksyntäkierron ja siirron ERP-järjestelmään. Tuotteen ollessa vain ostolaskujen kierrätysratkaisu, se ei pidä sisällään ERP-järjestelmässä tapahtuvia toimenpiteitä kuten maksatusta. Maksupalaute on mahdollista palauttaa takaisin ohjelmaan, kun maksatus on suoritettu ERP:ssä. Maksupalaute sisältää yleensä maksuviitteen ja maksupäivän. [Basware Purchase-to-Pay 2017.]



**Kuva 3.** Sähköinen ostolaskuprosessi ilman tilausta [Lahti ja Salminen 2014].

Asiakkaille tarjotaan mahdollisuus valita kolmesta standardista ostolaskuprosessista, jotka ovat otsikkotason hyväksyntä, tiliöinnin kustannuskohteisiin perustuva hyväksyntä, tiliöinnin kustannuskohteisiin perustuva tarkastus ja hyväksyntä. Tiliöinnin kustannuskohteita ovat esimerkiksi tilit ja kustannuspaikat. [Basware Purchase-to-Pay 2017.] Tässä tutkielmassa keskitytään erityisesti otsikkotasolla tapahtuvaan hyväksyntään.

### 3.3 Sähköinen ostolaskuprosessi tilaukseen perustuen

Sähköinen hankinta lähtee liikkeelle siitä, että yritys tarvitsee tiettyjä tuotteita tai palveluja toimintojensa ylläpitämiseen, kuten tavallisessa hankintaprosessissa. Hankintaehdotus luodaan, hyväksytään ja sen perusteella tehdään ostotilaus. Hyväksynnän jälkeen ostotilaus lähtee toimittajalle. Palvelun tai tuotteen saapuessa tapahtuu vastaanotto. Hankintaehdotus syötetään manuaalisesti järjestelmään. Hankintaehdotus voi syntyä myös automaattisesti perustuen esimerkiksi varastosaldoihin. Hankinta kattaa suorat hankinnat, jotka liittyvät oleellisesti yrityksen toimintaan. Lisäksi epäsuoria hankintoja voidaan tehdä erilaisten hankintaportaalien kautta. Epäsuorat hankinnat ovat tavaroita tai palveluita, jotka eivät liity suoraan yrityksen toimittamiin palveluihin tai tavaroihin asiakkailleen. [Lahti ja Salminen 2014.]

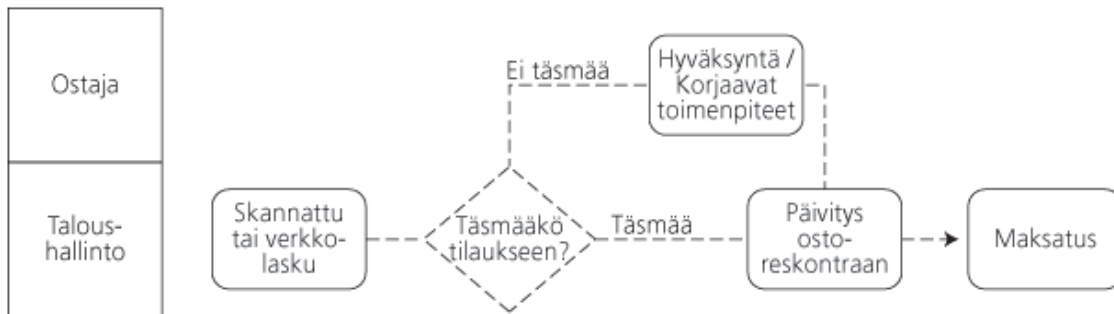
Baswaren P2P:ssa tilaaminen on aina automaattista. Käyttäjät eivät voi manuaalista luoda ostotilauksia. Järjestelmä luo ostotilauksen automaattisesti perustuen hankintaehdotukseen, joka on luotu käyttäjän toimesta ja lähettää sen toimittajalle hyväksynnän jälkeen ennalta määritellyllä tavalla. Yleisimpiä

tapoja ovat standardi XML muotoinen tilaus tai sähköposti, jossa ostotilaus on PDF-muodossa. Hankintaehdotuksen muuttuessa automaattisesti ostotilaukseksi yritys saavuttaa paremman automaation hankintaprosessissa. Hankintaprosessia nopeuttavat tämän lisäksi monet muut ominaisuudet. Yhdellä ostotilauksella on mahdollista tilata tuotteita monelta eri toimittajalta ja järjestelmä jakaa ostotilauksen automaattisesti osiin. Järjestelmään rakennetut validoinnit varmistavat, että ostotilauksella on kaikki tarvittavat tiedot. [Basware Purchase-to-Pay 2017.]

Tilauksen ja vastaanoton aikana tehty hyväksyntä- ja tiliöintitiedot mahdollistavat laskuautomaatiota. Tilauksella tallennettujen kustannuskohteiden hyödyntäminen ostolaskun saapuessa tehostaa prosessia huomattavasti. Tämä vaihe voidaan tehokkaimmillaan automatisoida niin, että tiedot saadaan automaattisesti täydennettyä ostolaskulle sen saapuessa järjestelmään. Ostolaskun saapuminen ei aiheuta enää manuaalisia toimenpiteitä. Täsmäytyksessä ostotilauksen tiedot täsmäävät ostolaskulle täytettäviin tietoihin ja lasku on valmis automaattisesti siirrettäväksi ostoreskontraan ja maksatukseen. [Lahti ja Salminen 2014.] Baswaren P2P-sovellus pystyy automatisoimaan ostotilauksen ja ostolaskun yhdistämisen toisiinsa perustuen ostotilauksen otsikko-, rivi- ja vastaanottotietoihin. Tietoja vertaillaan laskulla oleviin otsikko- tai rivitietoihin. Sovellus tarjoaa mahdollisuuden myös suunnittelemattomien lisäkulujen, kuten ylimääräisten rahtimaksujen automaattisen käsittelyyn. [Basware Purchase-to-Pay 2017.] Täsmäytystä käydään tarkemmin läpi teknisellä tasolla kohdassa 2.4.

Ostotilaukseen perustuva laskun käsittelyprosessi on seuraavanlainen (ks. kuva 4) [Lahti ja Salminen 2014]:

1. Järjestelmässä luodaan ostoehdotus ja hyväksytään. Luotu ostotilaus lähtee toimittajalle.
2. Toimittaja toimittaa tuotteen tai palvelun, ja ostotilaukselle tehdään vastaanotto.
3. Ostolasku saapuu järjestelmään, ja se pyritään täsmäyttämään automaattisesti ostotilaukseen.
4. Ostolaskulle tuodaan tiedot automaattisesti perustuen ostotilaukseen.
5. Laskun täsmätessä ostotilaukseen uutta hyväksyntäkiertoa ei tarvita. Mikäli laskulla on ostotilaukseen verrattuna esimerkiksi hintaeroa ennalta määriteltujen toleranssien yli, lasku lähtee hyväksyntäkiertoon. Tässä vaiheessa on mahdollista hyväksyä tai hylätä lasku, ja reklamoida toimittajalle.
6. Ostolasku siirretään ostoreskontraan ja maksetaan.



**Kuva 4.** Sähköinen ostolaskuprosessi tilaus [Lahti ja Salminen 2014].

### 3.4 Sähköinen ostolaskuprosessi sopimukseen perustuen

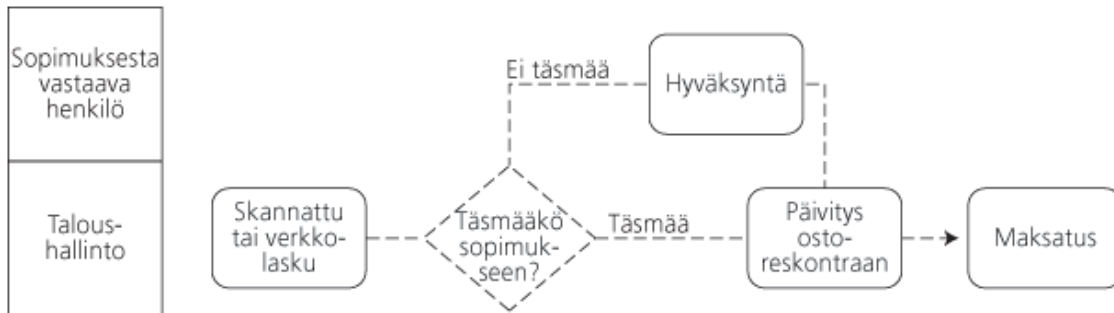
Sopimukseen perustuvan ostolaskun käsittely on hyvin samankaltainen, kuin ostotilaukseen perustuva ostolaskun käsittely. Ostolaskun täsmäytys tapahtuu sopimukseen ostotilauksen sijasta. Sopimukseen perustuviksi ostolaskuiksi yleensä määritellään erilaiset toistuvat laskut, kuten vuokrat ja autoleasing laskut. [Lahti ja Salminen 2014.]

Baswaren P2P tarjoaa kolmea erityyppistä sopimuksiin perustuvaa maksusuunnitelmaa. Ensimmäinen on aikatauluun perustuvat maksusuunnitelmat. Aikatauluun perustuvia maksusuunnitelmia käytetään useasti, kun toimittaja lähettää saman sisältöisen laskun toistuvasti tietyn ajan välein. Tästä on hyvänä esimerkkinä vuokralaskut, jotka yleensä saapuvat kerran kuussa samalla laskun loppusumalla, joka perustuu ostajan ja toimittajan väliseen vuokrasopimukseen. Maksusuunnitelmalle kirjataan kaikki tarvittavat yksityiskohdat, aikaväli ja odotettu laskun summa. Toisena vaihtoehtona on itse laskuttavat sopimukset, jotka luovat ostolaskun automaattisesti. Tämä on erityisesti hyödyllistä silloin, kun toimittaja ei lähetä erillistä laskua ostajalle, ja ostaja maksaa toimittajalle ennalta määritetyn maksuaikataulun ja sopimuksen mukaan. Kolmantena ja viimeisenä vaihtoehtona on budjettiin perustuvat maksusuunnitelmat. Toimittajalta tulevat laskut saapuvat epäsäännöllisin väliajoin perustuen kuitenkin ennalta määriteltyyn sopimukseen, esimerkiksi sovittuun projektitoimitukseen, joka sisältää erilaisia palveluita ja ostoja, joilla on ennalta määritetty budjetti kaikista menoista. [Basware Purchase-to-Pay 2017.]

Sopimukseen perustuva käsittelyprosessi jaetaan seuraaviin alakohtiin (ks. kuva 5) [Lahti ja Salminen 2014]:

1. Järjestelmässä luodaan sopimus, tiliöinti ja ostolaskun käsittelyyn tarvittavat säännöt, kuten tarkastajan ja hyväksyjän ennalta määrittäminen.
2. Ostolasku saapuu järjestelmään sopimusnumerolla.
3. Ostolaskulle tuodaan tiliöinti automaattisesti perustuen tehtyyn maksusuunnitelmaan.

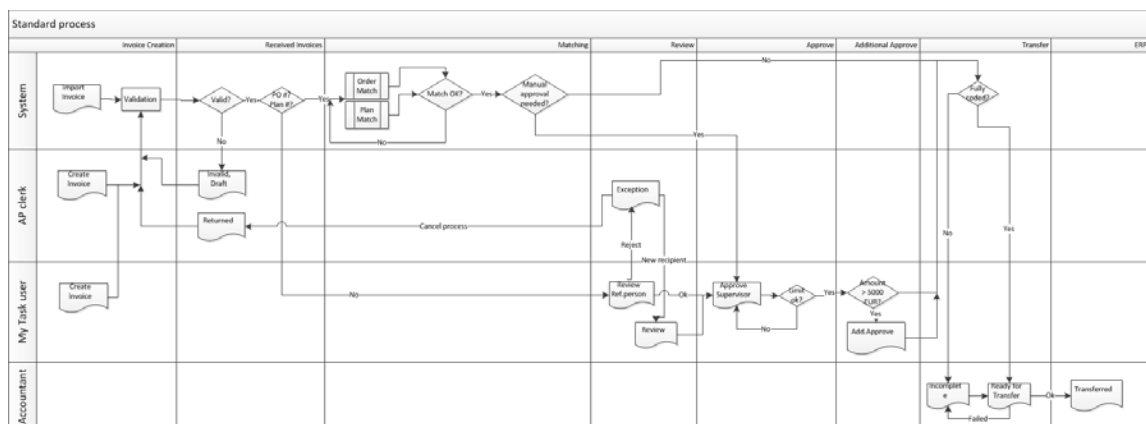
4. Laskun täsmätessä sopimukseen hyväksymiskiertoa ei tarvita, koska maksusuunnitelma on aikaisemmin hyväksytty. Eräpäivän, summan tai muiden tietojen ollessa erilaisia ostolasku lähtee hyväksyntäkiertoon.
5. Ostolasku siirretään ostoreskontraan ja maksetaan.



**Kuva 5.** Sähköinen ostolaskuprosessi sopimus [Lahti ja Salminen 2014].

### 3.5 Basware Purchase-to-Pay:n standardi sähköinen ostolaskuprosessi

Kuvassa 6 on kuvattu ostolaskuprosessin eri skenaariot Baswaren tarjoaman ratkaisun näkökulmasta. Tämä sisältää ilman sopimusta ja tilaukseen tai sopimukseen perustuvat ostolaskut, kuten kolmessa edellisessä kohdissa ja kuvissa 3, 4 ja 5 on esitelty. Standardiprosessi ei yleisellä tasolla poikkea paljoa siitä, mitä läpi käydyssä kirjallisuudessa on esitelty. Neljän silmän periaate toteutuu molemmissa prosesseissa. Taloushallinnon ja loppukäyttäjän väliset tehtävät on jaoteltu samanlaisiin tehtäviin eli tiliöintiin, tarkastukseen ja hyväksyntään. Taloushallinnon tehtäväksi jää useasti vain poikkeustapausten käsittely, ja loppukäyttäjät selviytyvät lopuista tehtävistä itsenäisesti. Yksi mainittava ero on, että lisähyväksyntä perustuu hyväksyjälle asetettuihin hyväksyntärajoihin. Jos hyväksyntäraja ei riitä ostolaskun hyväksyntään lähetetään ostolasku seuraavalla henkilölle lisähyväksyttäväksi, jolla on suurempi hyväksyntäraja. [Basware Purchase-to-Pay 2017.]



**Kuva 6.** Standardi otsikkotason hyväksyntä [Basware Purchase-to-Pay 2017].

## **5. Ostolaskuprosessin sähköistämisen keskeiset edellytykset**

Ostolaskuprosessin sähköistämisen keskeisin edellytys on, että kaikki ostolaskujen käsittelyyn vaadittavat tiedot ovat sähköisessä muodossa. Tärkeimpiä tietoja ovat laskulla vaadittavat perustiedot kuten toimittajat, kirjanpidontilit ja kustannuspaikat.

### **4.1 Perustietojen ylläpito**

Ostolaskuprosessille on tärkeää, että kaikki tarvittavat perustiedot ovat saatavilla, jotta prosessi sujuu mahdollisimman tehokkaasti. Perustietoja ovat esimerkiksi toimittajat, tilikartat ja kustannuspaikat, joista prosessille keskeisin tekijä on toimittajan perustiedot. Ostolaskun saapuessa toimittaja tunnistetaan, ja sen perusteella laskulle voidaan täydentää automaattisesti esimerkiksi toimittajarekisteriin asetettu pankkitili ja maksuehto, jolloin maksut saadaan tehtyä oikean sopimuksen ja valitun oletuspankkitilin mukaan. [Lahti ja Salminen 2014.]

Perustietojen ylläpito tapahtuu pääsääntöisesti ERP-järjestelmissä, joista tiedot siirretään säännöllisin väliajoin ostolaskujärjestelmään. Baswaren P2P:n mukana toimitettava anyERP-integraatioalusta mahdollistaa integraatiot ERP:n ja ostolaskujärjestelmän välillä. Integraatiotapoja ERP:stä riippuen ovat esimerkiksi tietyn skeeman mukaisten tiedostojen siirto SFTP-palvelimen kautta, ja eri tyyppiset web-rajapinnat kuten REST ja SOAP. [Basware Purchase-to-Pay 2017.]

Prosessin tehokkuuden kannalta on tärkeää, että laskulta pystymään automaattisesti poimimaan tiettyjä kohdistustietoja. Kohdistustietojen avulla lasku voidaan välittää automaattisesti tarkastettavaksi ja tehdä automaattista tiliöintiä ehdotusta perustuen esimerkiksi laskulla tulevaan kustannuspaikkaan. Kohdistustietojen taakse voidaan myös ylläpitää henkilöitä, joille laskuun liittyvän kustannuksen tarkastus tai hyväksyntä kuuluu. [Lahti ja Salminen 2014.] Verkkolaskurakenteessa on tietyt paikat, joihin tiedot tulisi laskun lähettävän toimittajan täydentää.

### **4.2 Ostolaskujen vastaanottotavat ja verkkolaskustandardit**

Ostolaskujärjestelmä voi vastaanottaa vain sähköisessä muodossa olevia laskuja, joten vastaanotto tapahtuu verkkolaskuna tai paperilaskuna skannauksen kautta. Skannauksessa pyritään poimimaan laskulta tarpeelliset tiedot, joita tyypillisesti ovat laskun päivä, eräpäivä, summa, viitenumero, valuutta, toimittajan tili, tilaus- tai sopimusnumerot. Verkkolaskutus poistaa skannauksessa esiintyvät manuaaliset työtehtävät ja suurimpia etuja ovat tietojen oikeellisuus ja prosessin nopeutuminen. Erilaiset taloushallinnon sovellukset ja ERP-järjestelmät tarjoavat ja mahdollistavat verkkolaskujen lähetyksen ja vastaan-

oton. [Lahti ja Salminen 2014.] Basware on yksi verkkolaskutusta tarjoavista operaattoreista, jotka hoitavat laskujen skannausta ja välittävät laskudataa yritysten välillä eli toimivat verkkolaskuoperaattorina [Basware Purchase-to-Pay 2017].

Verkkolaskulla data on ennalta määritellyn sanomakuvauksen mukaista, yleensä XML-muotoista [Spanic *et al.* 2011]. Datan lisäksi laskulla säilytetään laskun kuva erityisesti kierrätystä ja arkistointia varten. Sähköinen lasku vähentää manuaalisten tehtävien määrää ja mahdollistaa automaation rakentamisen, kun tiedot voidaan tuoda suoraan ostolaskujärjestelmään. [Lahti ja Salminen 2014.] Maailmalla on olemassa lukuisia eri XML-pohjaisia standardeja verkkolaskujen lähetykseen. Standardien suuri määrä asettaa haasteita laajemmalle yleisen yhden standardin käyttöönotolle. [Spanic *et al.* 2011.] Verkkolaskutusta Euroopan tasolla edistää EESPA (European E-invoicing Service Providers Association), joka pyrkii omalla toiminnallaan edistämään yhteisen standardin löytämistä, ja EU-lainsäädäntöjen esteiden poistamista [Lahti ja Salminen 2014]. Tunnetuimmaksi standardiksi määritellään OSASIS Universal Business Language (UBL 2.0). [Spanic *et al.* 2011.] Suomessa käytettävät verkkolaskustandardit ovat Tiedon ja Aditron omistava TEAPPS-sanomakuvaus [Tieto.fi 2017] ja Finanssiala ry:n ylläpitämä Finvoice-verkkolaskukuvaus [Finanssiala.fi 2017].

#### **4.3 Ostolaskujen tiliöinti, tarkastus ja hyväksyntä**

Ostolaskut lähtevät automaattisesti kiertoon riippuen käsittelyjärjestelmään asetetuista säännöistä. Kiertoon lähetys voi olla myös manuaalinen ostoreskontran työtehtävä, jos laskulle halutaan tehdä kiertoa edeltävää tiedon tarkistusta. Kiertoon lähtiessä tiliöinti tehdään pääsääntöisesti täsmätyn ostotilauksen tai sopimuksen taakse asetettujen tiliöintien mukaan. Kululaskuille eli ei ostotilaukseen tai sopimukseen perustuvilla laskuilla automaatiota voidaan rakentaa, jos samalta toimittajalta vastaanotetaan usein saman sisältöisiä laskuja. Oletustiliöinnin asettaminen tietylle toimittajalle parantaa tarkkuutta, ja poistaa manuaalista työtä. Tiliöinti on mahdollista poimia suoraan toimittajan lähettämältä verkkolaskulta. [Lahti ja Salminen 2014.] Kululaskuille voidaan tiliöintiä automatisoida myös perustuen aikaisempien mallipohjien, automaattiosääntöjen ja historian perusteella [Basware Purchase-to-Pay 2017]. Jos ei ole mahdollisuutta automaattiseen tiliöintiin, yrityksestä riippuen se tehdään ostoreskontran tai tarkastajan toimesta. Ostoreskontran tekemää tiliöintiä pidetään tehokkuuden ja oikeellisuuden näkökulmasta vahvempänä vaihtoehtona. [Lahti ja Salminen 2014.]

Ostolaskuilla pätee kaksiportainen periaate. Kahden käsittelijän periaatteella tarkoitetaan, että laskulla on kierron aikana tarkastus sekä hyväksyntä

[Ahokas 2012]. Esimerkiksi tilaaja on laskun tarkastaja ja tilaajan esimies on laskun hyväksyjä [Lahti ja Salminen 2014]. Käyttäjän taakse voidaan P2P ostolaskujärjestelmässä asettaa esimies, joka toimii laskun oletushyväksyjänä. Tarkastajaa pyritään etsimään laskulla olevien viitetietojen esimerkiksi viitehenkilön tai kustannuspaikan perusteella, lisäksi hyväksyjää voidaan ehdottaa ja hakea automaattisesti perustuen esimerkiksi toimittajaan tai tiliöinnin kustannuskohteisiin. [Basware Purchase-to-Pay 2017.] Käyttäjille asetetaan organisaation hyväksyntäpolitiikan mukaiset rajat ja hyväksyntäoikeudet, jotka ovat linjassa yrityksen organisaatiorakenteen mukaisesti. Sopimukseen perustuvat laskut voidaan automatisoida, eikä erillistä tarkastusta ja hyväksyntää tarvita, koska hyväksyntä on tehty sopimuksen luontivaiheessa. Mikäli lasku ei automaattisesti täsmää sopimukseen esimerkiksi puuttuvan sopimusnumeron tai virheellisten summien takia, lasku lähetetään kiertoon sopimukseen asetetulle oletustarkastajalle. Ostotilaukseen perustuvat laskut voidaan myös hyväksyä luontivaiheessa. Tilauksellisten laskujen täsmäytys vaatii, että vastaanotetut määrät ja summat vastaavat ostotilauksen ja laskun välillä. Täsmäytys on usein automatisoitu ostolaskujärjestelmän toimesta, mikäli automaattinen täsmäytys ei onnistu lasku voidaan täsmäyttää manuaalisesti, ja lähettää tarvittaessa kiertoon hyväksyjälle. [Lahti ja Salminen 2014.]

Täsmäytys tapahtuu eri tavalla riippuen siitä, onko ostotilaus vastaanottoon perustuva vai ei. Ostotilauksen ollessa vastaanottoon perustuva tulee käyttäjän tehdä tavaran tai palvelun vastaanotto, jotta täsmäytys on mahdollista. Vastaanottoon perustuvassa täsmäytyksessä vertailua tehdään vastaanotettujen määrien ja hintojen perusteella, eli ostolaskun määrää ja hintaa verrataan ostotilauksella vastaanotettuihin määriin ja hintoihin. Jos ostotilaus ei perustu vastaanottoon, vertailu tehdään ostotilausrivillä olevien hintojen ja määrien perusteella. Täsmäytys tapahtuu laskulla olevien rivitietojen, kuten ostotilausnumeron, läheteiden ja tuotetunnuksien perusteella. Jos rivitietoja ei ole saatavilla tai ne ovat muuten puutteellisia, täsmäytystä voidaan automaattisesti tehdä summien ja vastaanotettujen määrien perusteella. P2P:ssa tätä kutsutaan best-fit-täsmäytykseksi. [Basware Purchase-to-Pay 2017.]

#### **4.4 Ostolaskujen siirto kirjanpitoon ja maksatukseen**

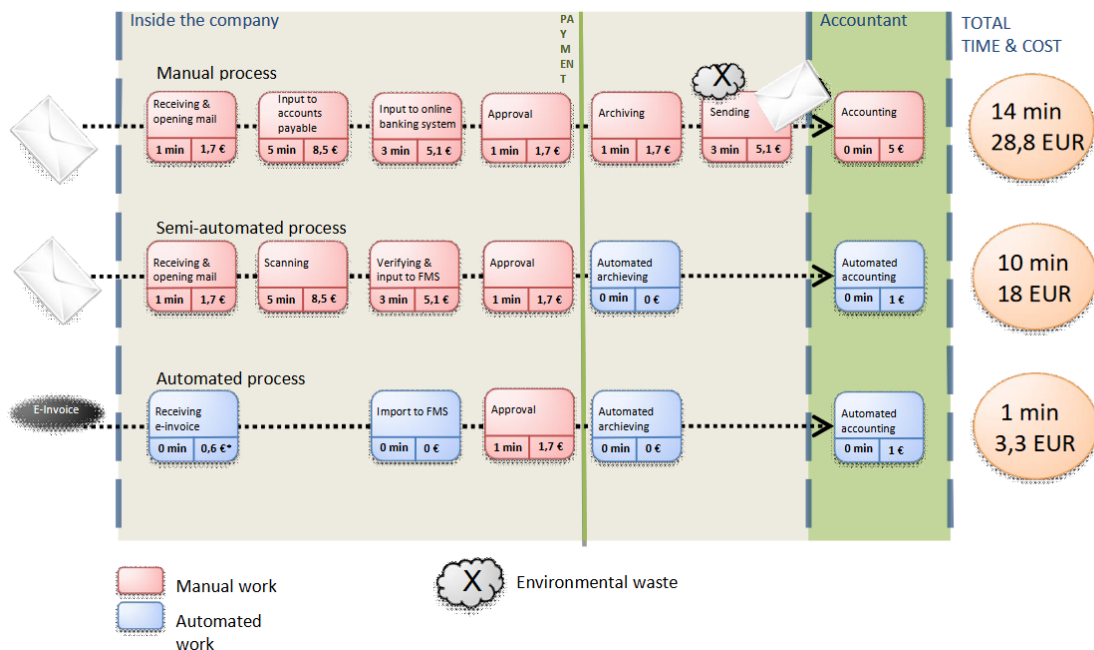
Ostolaskut ovat siirtovalmiina hyväksyttyjä ja suorittaneet niille asetetun kierroksen ja toimenpiteet ostolaskujärjestelmässä. Ostolaskujen siirto kirjanpitoon tapahtuu ostolaskujärjestelmän ja kirjanpitojärjestelmän välisellä integraatiolla, joka on yleensä toteutettu samoilla metodeilla kuin perustietojen integraatiot. Siirtotehtävä voi olla joko automaattinen tai manuaalinen toimenpide, automaation kannalta on oleellista, että ainakin osa laskuista voidaan automaattisesti siirtää kirjanpitoon. [Lahti ja Salminen 2014.]

Ostolaskujen maksaminen kirjanpitojärjestelmässä tehdään yrityksestä riippuen päivittäin tai viikoittain. Ajan ja helpomman kassanhallinnan saavuttamiseksi on suositeltavaa suorittaa maksatus 1-2 kertaa viikossa [Lahti ja Salminen 2014]. Vaikka ostolaskujärjestelmän laskujen käsittelyprosessi päättyy siinä vaiheessa, kun lasku on kirjattu kirjanpitoon, on tyypillistä, että maksatuksesta palautetaan ainakin maksupäivä ja maksuviite takaisin järjestelmään, jotka tallennetaan laskun otsikkotietoihin varattuihin kenttiin.

## 6. Ostolaskuautomaation mitatut hyödyt

Helsingin kauppakorkeakoulun 2008 tekemässä tutkimuksessa on mitattu, kuinka yritykset voivat saavuttaa aika- ja kustannussäästöjä automaattisella laskun vastaanotolla. Kuvassa 7 on esitelty laskujen vastaanottaminen yrityksessä, käsittelyajat ja mitatut kustannussäästöt. [Penttinen 2008.]

Kuvasta 7 voidaan päätellä, että Lahden ja Salmisen [2014] esittämä tieto pitää paikkaansa siitä, että ostolaskujen sähköisen vastaanoton ja käsittelyn on havaittu vaikuttavan kaikkein eniten laskun käsittelyn automaation rakentamiseen ja kustannuksiin. Puutteelliset laskutiedot ja paperilaskut ovat yritykselle kalliita. Täysin manuaalisen ja automaattisen prosessin välillä on huomattava aika- ja euromääräinen ero.



**Kuva 7.** Laskujen vastaanotto mikoyrityksessä [Penttinen 2008].

## 7. Sähköisen ostolaskujärjestelmän tulevaisuus

Ostolaskujen käsittelyn ennustetaan lähivuosina keskittyvän ostolaskuprosessin tehokkuuden ja automaation parantamiseen. Yrityksille on tärkeää, että



prosessit ja ohjeistukset ostolaskujen käsittelystä ovat kunnossa koko organisaatiossa. Automatiikka on saatu parannettua ostotilaukseen ja sopimuksiin perustuvien laskujen kanssa, epäsuorien hankintojen eli kululaskujen automatisointi on edelleen hankalaa, ja yritykset tulevat jatkossakin keskittymään enemmän tilauksellisten laskujen automaatioon. [Lahti ja Salminen 2014.] Pyrkimyksenä on kuitenkin kehittää myös kululaskujen automaatiota silloin, kun lasku ei perustu ostotilaukseen tai sopimukseen. Verkkolaskudatassa tulevan tiedon perusteella tehtävät automaattiset tiliöinnit, ja aikaisempiin saman toimittajan laskuihin perustuvien tiliöintiehdotuksien tarjoaminen ovat tällä hetkellä esillä olevia esimerkkejä [Lahti ja Salminen 2014, Basware.com 2017]

## 8. Johtopäätökset ja yhteenveto

Yrityksille on tärkeää saavuttaa parannuksia kustannuksissa ja tehokkuuksissa prosesseissa. Ostolaskujen käsittelyn automaation mahdollisuuksien ja koko hankinnasta maksuun prosessin ymmärtäminen säästöjen saavuttamiseksi on tärkeitä. Tämä tutkielma esitteli mahdollisimman laajasti ostolaskuprosessin ja sen tarjoamia mahdollisuuksia. Tekoälyn ja robotiikan kehittyessä ostolaskujen käsittely ottaa varmasti valtavia askelia eteenpäin automaation näkökulmasta tarkasteltuna, mutta myös nykyisillä tutkielmassa käsitellyillä prosesseilla on mahdollista saavuttaa automaatiotasosta riippuen mittavia tehokkuus- ja kustannussäästöjä.

Osoittautui, että tieteellistä kirjallisuutta ja tehtyä tutkimusta löytyy hyvin niukasti ostolaskujen käsittelystä ja ostolaskujärjestelmästä, joten tutkielma nojautui ostolaskujärjestelmää käsittelevässä kokonaisuudessa suurimmalta osin Baswaren tuotekuvauksiin [2017] ja Lahtisen ja Salminen digitaalisen taloushallinnon teokseen [2014]. Tämä teos oli ainoa, josta löytyi sähköistä ostolaskuprosessia kuvaavaa ja analysoivaa tietoa. Verkkolaskutuksesta löytyy huomattavasti tarkemmalla tasolla tutkittua tietoa, tilastoja ja monipuolisempia lähteitä. Syvemmän analyysin tekeminen oli näin ollen helpompaa. Tämä kertoo siitä, että verkkolasku on saavuttanut suuremman mielenkiinnon ja markkinat.

## Viiteluettelo

- Hamid Reza Ahadi. 2004. A comparative study of the application of electronic data interchange and internet technology to business process reengineering. *Tsinghua Science and Technology. J. IEEE* 9, 4, 489-496.
- Niina Ahokas. 2012. *Yrityksen sisäinen valvonta*. Edita.
- Basware. 2017. Purchase-to-pay feature timeline. <http://releases.basware.com/timeline>. Viitattu 3.10.2017.

- Basware. 2017. Purchase-to-Pay Solution Reference Guide Version 17.5.
- Eduardo B. Fernandez and Xiaohong Yuan. 2009. An analysis pattern for invoice processing. In: *Proc. of the 16th Conference on Pattern Languages of Programs*. 1-10.
- Finanssialan keskusliitto. 2017. Finvoice verkkolasku yrityksille. <http://www.finanssiala.fi/finvoice/Sivut/default.aspx>. Viitattu 8.11.2017.
- Jonne Hellgren ja Maija Tenhunen. 2010. Ympäristöystävällinen verkkolasku. Finanssialan keskusliitto. [http://www.finanssiala.fi/materiaalit/Ymparistoystavallinen\\_verkkolasku.pdf](http://www.finanssiala.fi/materiaalit/Ymparistoystavallinen_verkkolasku.pdf). Viitattu 21.11.2017.
- Bruno Koch. 2017. E-invoicing / E-billing. Billentis. <https://www.basware.com/fi-fi/ratkaisut> (<https://resource.basware.com/c/e-invoicing-e-billing>). Viitattu 7.10.2017
- Sanna Lahti ja Tero Salminen. 2014. *Digitaalinen taloushallinto*. Talentum.
- Hsien-Yu Lee and Nai-Jian Wang. 2013. The implementation of integrating e-procurement, e-contracting and e-invoice platforms for the B2B E-MarketPlace web-based system. *International Journal of Engineering Research* 2, 4, 300-309.
- Michael Netter. Eduardo B Fernandez. Gunther Pernul. 2010. Refining the pattern-based reference model for electronic invoices by incorporating threats. In: *Proc. of International Conference on Availability, Reliability and Security*. 560-564.
- Esko Penttinen. 2008. Electronic invoicing initiatives in Finland and in the European Union – Taking the steps towards the real-time economy. Helsingin kauppakorkeakoulun julkaisuja B-95. <http://epub.lib.aalto.fi/pdf/hseother/b95.pdf>. Viitattu 22.11.2017.
- Davor Spanic, Denis Ristic, Boris Vrdoljak. 2011. An electronic invoicing system. In: *Proc. of the 11th International Conference on Telecommunications*. 149-156.
- Widjojo Suprpto, Zeplin Jiwa Husada Tarigan, Sautma Ronni Basana. 2017. The influence of ERP system to the company performance seen through innovation process, information quality, and information sharing as the intervening variables. In: *Proc. of the 2017 International Conference on Education and Multimedia Technology*. 87-91.
- Tieto. 2017. TEAPPSXML-versio 2.7.2. <https://www.tieto.fi/palvelut/liiketoimintaprosessienhallinta/business-information-exchange/laskuhotelli-palvelun-ohjeita-jakuvauksia/teappsxml-versio-272>. Viitattu 8.11.2017.

- Tilastokeskus / Suomen virallinen tilasto (SVT). 2013. Tietotekniikan käyttö yrityksissä, liitetaulukko 7: Sähköinen lasku vuonna 2013, osuus kaikista luokan yrityksistä. [http://www.stat.fi/til/ichte/2013/ichte\\_2013\\_2013-11-26\\_tau\\_007\\_fi.html](http://www.stat.fi/til/ichte/2013/ichte_2013_2013-11-26_tau_007_fi.html). Viitattu 21.11.2017.
- Lucie Vesela and Miroslav Radimersky. 2014. The development of electronic document exchange. *Procedia Economics and Finance* 12 (Supplement C), 743-751.
- Yrittäjät. 2014. Sähköinen taloushallinto. <https://www.yrittajat.fi/yrittajan-abc/taloushallinto-ja-maksut/taloushallinto/sahkoinen-taloushallinto-317818>. Viitattu 28.9.2017.

# Lisätyn todellisuuden sovellusten käyttömahdollisuuksia neurokirurgisissa toimenpiteissä

**Emma Hanhiniemi**

## **Tiivistelmä.**

Neurokirurgia on ensimmäisiä lääketieteen aloja, joka alkoi tutkia, miten lisätyn todellisuuden sovelluksia voitaisiin hyödyntää toimenpiteissä. Neurokirurgisten toimenpiteiden tunkeutuvuutta pyritään jatkuvasti vähentämään potilasturvallisuuden parantamiseksi. Tunkeutuvuuden vähentämisellä tarkoitetaan tässä kirurgisissa toimenpiteissä muun muassa leikkausviiltojen pienentämistä sekä avoleikkausten osuuden vähentämistä. Tunkeutuvuuden vähentäminen on kuitenkin tehnyt leikkausalueista ahtaita, ja näkyvyys niissä on usein heikko. Lisätyn todellisuuden sovelluksilla onkin suuri potentiaali parantaa leikkausalueiden näkyvyyttä potilaan päälle projisoidun tietokonegrafiikan avulla.

Tutkielma on kirjallisuuskatsaus, jossa perehdytään erityisesti viime vuosina julkaistuihin artikkeleihin ja tutkimuksiin lisätyn todellisuuden sovellusten hyödyntämisestä neurokirurgisissa toimenpiteissä. Tutkielmassa käydään läpi tällä hetkellä käytössä olevia neurokirurgisten toimenpiteiden paikannusmenetelmiä sekä niiden haasteita, joita lisätyn todellisuuden sovellusten avulla voidaan ratkaista. Tarkoituksena on tehdä katsaus lisätyn todellisuuden sovellusten tutkimuksen nykytilaan neurokirurgiassa sekä perehtyä sen tuomiin hyötyihin. Tutkielmassa käydään myös läpi lisätyn todellisuuden sovelluksiin liittyviä haasteita ja niiden mahdollisia ratkaisuja tulevaisuudessa.

**Asiasanat ja -sanonnat:** lisätty todellisuus, laajennettu todellisuus, neurokirurgia, neuronavigaatio

## **1. Johdanto**

Kirurgisten, erityisesti neurokirurgisten toimenpiteiden invasiivisuutta eli tunkeutuvuutta pyritään jatkuvasti vähentämään. Jokainen kirurginen toimenpide on itsessään väistämättä invasiivisin, mutta itse toimenpiteen invasiivisuuden väheneminen parantaa potilasturvallisuutta. Potilasturvallisuus parantuu pääosin, kun leikkausaluetta ympäröivän terveen kudoksen vaurioiden riski vähenee [Meola *et al.* 2016]. Invasiivisuuden väheneminen leikkauksissa kuitenkin pienentää myös toimenpidettä suorittavan kirurgin näkökenttää leikkausalueella, ja lisää siten leikkauksen vaikeutta.

Neurokirurgiset toimenpiteet nojaavat vahvasti aivoista kuvattuun materiaaliin sekä invasiivisuuden vähenemisen että aivojen fysiologian vuoksi [Tagaytayan *et al.* 2016]. Tällä hetkellä potilaasta kuvattu materiaali esitetään leikkauksalin näytöillä toimenpiteen aikana. Osa kuvista on kaksiulotteisia ja osa

niiden pohjalta tehtyjä kolmiulotteisia mallinnuksia. Koska kuvat esitetään kuitenkin näytöillä, näkee kirurgi kuvat ja mallinnukset pohjimmiltaan kaksiulotteisina. Kirurgin vastuulle jääkin leikkaustilanteessa muuntaa kuvat mielessään kaksiulotteisista kolmiulotteisiksi ja soveltaa niitä oikeassa asennossa ja koossa potilaan aivoihin nähden [Pandya *et al.* 2005]. Lisätyn todellisuuden sovelluksilla voidaan parantaa näkyvyyttä potilaan päälle generoidulla tietokonegrafiikalla. Grafiikalla voidaan myös havainnollistaa toimenpidettä suorittavalle kirurgille potilaan anatomiaa paremmin kuin nykyisin käytössä olevilla menetelmillä. Lisäksi lisätyn todellisuuden sovellusten avulla voidaan helpottaa kirurgien työergonomiaa sekä työnkulkua [Inoue *et al.* 2013; Tagaytayan *et al.* 2016].

Tutkielman luvussa 2 perehdytään neuronavigaatioon, joka on kokoelma menetelmiä, joiden avulla neurokirurgisissa toimenpiteissä paikannetaan leesiöitä. (Leesiolla tarkoitetaan vammaa, vauriota, vioittumaa, häiriötä; elimen rakenteen tai toiminnan sairaalloista muutosta [Duodecim 2017].) Tutkielman luvussa 3 esitellään lyhyesti menetelmiä, joita neuronavigaatioissa tällä hetkellä hyödynnetään sekä niitä puutteita, joita lisätyn todellisuuden sovelluksilla pyritään parantamaan. Luvussa 4 paneudutaan lisättyyn todellisuuteen osana keinotodellisuuksien kirjoa, ja käsitellään sen ominaispiirteitä. Tutkielman luku 5 käsittelee sitä, miten neurokirurgisissa toimenpiteissä on hyödynnetty lisätyn todellisuuden sovelluksia, sen mahdollisuuksia ja ongelmia. Tutkielman viimeinen luku on yhteenveto, jossa kerrataan tutkielmassa läpikäytyjä aiheita sekä pohditaan, mitä tulevaisuus tuo alalle mukanaan.

## 2. Neuronavigaatio

Moderni neurokirurgia on pyrkinyt jo 1800-luvun lopulta kehittämään tarkkoja paikannusmenetelmiä vähentääkseen toimenpiteiden invasiivisuutta, ja siten parantamaan potilasturvallisuutta sekä vähentämään potilaskuolleisuutta [Meola *et al.* 2016]

Neuronavigaatio on neurokirurgiassa 1900-luvun alusta asti käytössä ollut kokoelma menetelmiä, joita käytetään leesioiden paikannukseen aivoissa. Neuronavigaation menetelmiä on erilaisia, mutta niiden kaikkien periaate on sama: ennen toimenpidettä potilaan aivoista otetaan kuvia, joiden perusteella leesion tai muun häiriön sijainti voidaan paikantaa aivoissa. Toimenpiteessä potilaan pään alue kohdistetaan (engl. register) kuvien kanssa niin, että kuvia voidaan käyttää luotettavana lähteenä. Kohdistamisessa potilaan pää asetetaan siis samaan linjaan kuvien kanssa. Myös toimenpiteen aikana potilaan aivoista otetaan kuvamateriaalia, ja se yhdistetään ennen operaatiota saatuun kuvamateriaaliin mahdollisimman tarkan kokonaiskuvan saamiseksi. Kuvat esitetään leikkaussalin näytöllä, josta kirurgi voi käyttää niitä viitteenä suorittaessaan toimenpidettä. Neuronavigaatiota voidaankin verrata esimerkiksi autoissa käy-

tettäviin GPS-laitteisiin, joita ajaja ei välttämättä tarvitse ajaessaan tuntemallaan alueella, mutta joka on erittäin hyödyllinen alueella, joka ei ole kovinkaan tuttu. [Ivanov and Ciurea 2009]

Neuronavigaatiossa käytetty teknologia ja menetelmät ovat kehittyneet merkittävästi vuosien saatossa. 1800-luvun lopussa ja 1900-luvun alussa leesioiden paikantaminen perustui miltei täysin potilaan oireiden perusteella tehtyihin päätelmiin [Ivanov and Ciurea 2009], sillä tuolloin ei ollut saatavilla teknologiaa, jonka avulla aivoista olisi saatu kuvamateriaalia.

Dandy [1918] käsitteli ensimmäisen kerran aivojen kuvaamiseen onnistuneesti käytettyä *ventrikulografiaa*. Tämä voidaankin laskea ensimmäiseksi neuronavigaation menetelmäksi. Aivojen ventrikulografiassa aivoista valutetaan aivo-selkäydinneste pois, ja sen tilalle puhalletaan ilmaa, jonka avulla aivojen rakenne näkyy röntgenkuvissa selkeämmin. Vuodesta 1927 lähtien aivojen kuvaaminen helpottui *angiografian* kehityksen myötä, jossa aivoihin ilman puhaltamisen sijaan potilaan verisuoniin ruiskutetaan varjoainetta, joka saa verisuonet erottumaan röntgenkuvissa tarkasti.

1970-luvulta lähtien aivojen kuvaamiseen on käytetty CT-kuvaamista eli aivojen tomografiaa [Ivanov ja Ciurea 2009]. Tomografiassa käytetään röntgensäteitä, mutta yhden kuvan sijaan tomografiassa otetaan lyhyen ajan välein monta kuvaa aivoista kerroksittain. Tästä syystä aivojen tomografiaa kutsutaan myös nimellä tietokonekerroskuvaus. Saaduista kuvista muodostetaan kolmiulotteinen mallinnus aivoista.

Toinen hallitseva aivojen kuvausteknologia on MRI- eli magneettikuvaus, joka on yleistynyt 1970–1980-luvuilta lähtien. Magneettikuvauksella pystytään tarkastelemaan aivokudosta toisin kuin röntgenkuvilla, joissa ilman varjoaineita pystytään tarkastelemaan vain kallon rakennetta. Magneettikuvaus on yleinen aivojen kuvaustapa myös siksi, että siinä ei synny ionisoivaa säteilyä, eikä se niin ollen aiheuta potilaalle terveydellistä haittaa. [Mustajoki ja Kaukua 2008.]

Kuvausmenetelmien lisäksi neuronavigaation tukena käytetään anatomisia menetelmiä. Anatomisilla menetelmillä tarkoitetaan tässä neurokirurgiassa käytettyjä navigointimenetelmiä, jotka perustuvat aivojen anatomiaan, kuten verisuonten sijaintiin sekä kallon muotoihin, ja niistä tutkittuun tietoon. Anatomista dataa on käytetty neuronavigaatiossa sekä ennen että jälkeen MRI- ja CT-kuvausmenetelmien yleistymisen. [Ivanov ja Ciurea 2009.]

Neuronavigaatio on hyödyllinen kokoelma menetelmiä, jonka suurin etu on neurokirurgisten toimenpiteiden invasiivisuuden vähentäminen ja sen myötä potilasturvallisuuden parantaminen.

### 3. Neuronavigaation menetelmät

Neuronavigaation menetelmät voidaan jakaa karkeasti kahteen ryhmään: funktionaalisiin sekä anatomisiin menetelmiin. Funktioneaalisiin menetelmiin laskeaan aivoista kuvaamalla saadun datan hyödyntäminen, sillä tämän datan avulla kirurgit näkevät, millä aivojen funktionaalisella alueella leesio sijaitsee [Ivanov and Ciurea 2009]. Aivoja voidaan kuvata erilaisin tavoin, mutta suosituimmat menetelmät siihen ovat magneettikuvaus sekä aivojen tomografia.

MRI- sekä CT-kuvaus muodostavat nykyaikaisen neuronavigaation peruspilarit. Kaikissa tilanteissa näitä kuvaustapoja ei voida kuitenkaan käyttää. Esimerkiksi magneettikuvauslaitteen magneetti on niin voimakas, että se tuhoaa esimerkiksi sydämentahdistimen, ja CT-kuvassa potilaan kehossa mahdollisesti olevat metallikappaleet, kuten implantit tai hampaiden paikat, voivat peittää leesioiden näkyvyyttä. Tällaisissa tilanteissa kuvaamiseen voidaan käyttää myös muita menetelmiä. Kuvat otetaan ennen neurokirurgisia toimenpiteitä, ja kuvia käytetään sekä sellaisenaan että niiden avulla tehdään aivoista myös usein kolmiulotteinen mallinnus, jonka avulla kirurgien on helpompaa muodostaa kokonaiskuva aivoista. Toimenpiteen aikana kuvat sekä mallinnukset esitetään leikkaussalin näytöillä, jotta kirurgi voi käyttää niitä viitteenä.

Kuvausmenetelmien lisäksi neuronavigaation apuna käytetään anatomisia paikannusmenetelmiä, jotka perustuvat pään verisuonten, hermoston ja kallon muotojen, kuten lovien, kohoumien ja urien avulla aivojen eri alueiden määrittelyyn. Anatomisia menetelmiä hyödynnetään neurokirurgisen toimenpiteen aikana: ensin kirurgi paikantaa oikean leikkausalueen kallon muodon sekä muiden ulkoisten merkkien perusteella. Kallon avaamisen jälkeen toimenpidettä suorittava kirurgi käyttää viitteenä muun muassa potilaan verisuonia, hermostoa sekä aivokudoksen poimuja. Kirurgit pitävät sekä funktionaalisen että anatomisen datan yhdistelmää parhaana lähestymistapana neuronavigaatioon. [Ivanov and Ciurea 2009.]

Nykyisissä neuronavigaation menetelmissä suurimmat puutteet liittyvät kirurgin työnkulun häiriintymiseen ja työergonomian haasteisiin. Magneettikuvien ja aivojen tomografian avulla saatu data on luotettavaa, ja sekä itse kuvat että niiden pohjalta luodut kolmiulotteiset mallinnukset aivoista muodostavat neuronavigaation kivijalan. Koska kuvat ja mallinnukset esitetään kuitenkin erillisillä näytöillä toimenpiteen aikana, tulee kirurgin siirtää katseensa leikkausalueelta näytöille ja takaisin lyhyen ajan välein. Erilaiset kuvat esitetään omilla näytöillään. Tämä jatkuva vilkuilu potilaan ja näyttöjen välillä luo häiriöitä kirurgin työnkulkuun. [Tagaytayan *et al.* 2016; Inoue *et al.* 2013.]

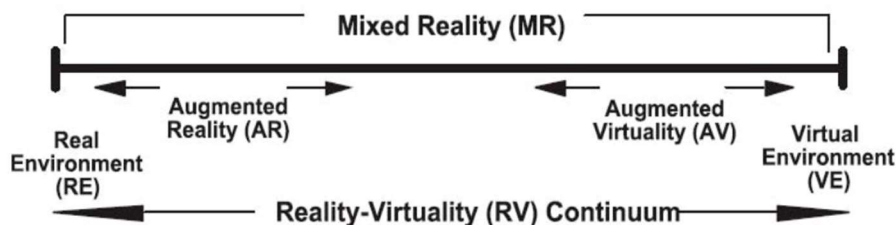
Työergonomiaan liittyvien ongelmien lisäksi suuri haaste leikkauksissa on kuvien ja mallinnusten soveltaminen leikkausalueelle. Koska kuvat esitetään näytöillä ja ne ovat siis pohjimmiltaan kaksiulotteisia, tulee kirurgin kyetä so-

veltamaan kuvien sisältö leikkausalueelle. Tämä vaatii, että kirurgi muuntaa kaksiulotteiset kuvat mielessään kolmiulotteisiksi, ja soveltaa niitä oikeassa koossa sekä orientaatiossa leikkausalueelle [Inoue *et al.* 2013]. Toimenpidettä suorittavan kirurgin tulee siis katsottuaan kuvaa pystyä siirtämään sen sisältö leikkausalueelle, joka on erillään kuvasta [Meola *et al.* 2016]. Potilaan orientaatio on tyypillisesti myös eri, kuin kuvien ja mallinnusten. Kuvat ja mallinnukset eivät myöskään reagoi kirurgin liikkeeseen interaktiivisesti.

Neurokirurgian invasiivisuuden vähentymisen vuoksi leikkausalueet ovat usein ahtaita ja pieniä, ja näkyvyys on heikko. Kirurgin täytyykin nykypäivänä kehittää itselleen ns. röntgenkatse, jolla hän pystyy navigoimaan leikkauksessa aivojen anatomisten indikaattorien, kuten esimerkiksi verisuonten ja hermojen perusteella vähentääkseen leesiota ympäröivien kudosten vaurioittamista. [Meola *et al.* 2016.]

#### 4. Lisätty todellisuus

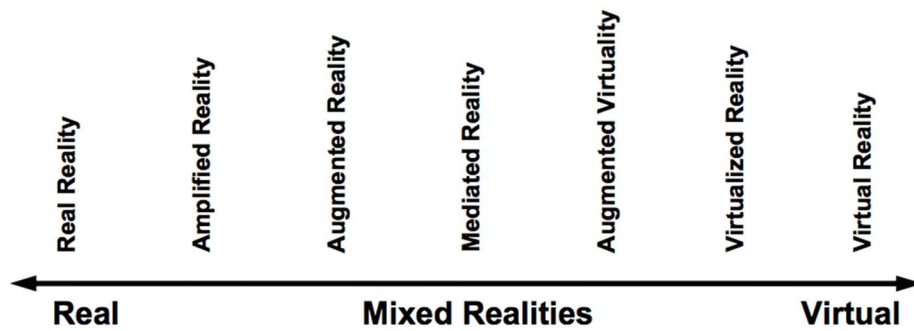
Lisätty todellisuus eli AR (engl. augmented reality) on yksi keinotodellisuuden muoto. Keinotodellisuuksista puhuttaessa monen ensimmäinen mielikuva liittyy virtuaalitodellisuuden sovelluksiin, joissa käyttäjä on täysin uppoutunut virtuaalimaailmaan. Virtuaalitodellisuuden ominaispiirre onkin, että todellinen todellisuus korvataan täysin virtuaalisella, ja käyttäjän ollessa vuorovaikutuksessa virtuaalitodellisuuden kanssa hän ei ole vuorovaikutuksessa todellisen maailman kanssa.



Kuva 1: Milgramin [1994] jatkumo.

Lisätty todellisuus sijoittuu sekä Milgramin ja Fushikinon [1994] että Schnabelin ja muiden [2007] jatkumoilla lähemmäs todellista maailmaa (tai reaalitytodellisuutta). Lisätyssä todellisuudessa käyttäjä näkee todellisen ympäristönsä, ja ympäristöä tehostetaan lisäämällä siihen tietokoneella luotua grafiikkaa. Lisätyssä todellisuudessa todellista ympäristöä ei siis korvata virtuaalisella sisällöllä, vaan siihen lisätään virtuaalisia elementtejä. Tästä tulee termi lisätty (toisinaan käytetty myös laajennettu) todellisuus. Ideaalitilanteessa käyttäjä ei erota todellisia ja virtuaalisia elementtejä toisistaan. [Milgram and Fushikino 1994]





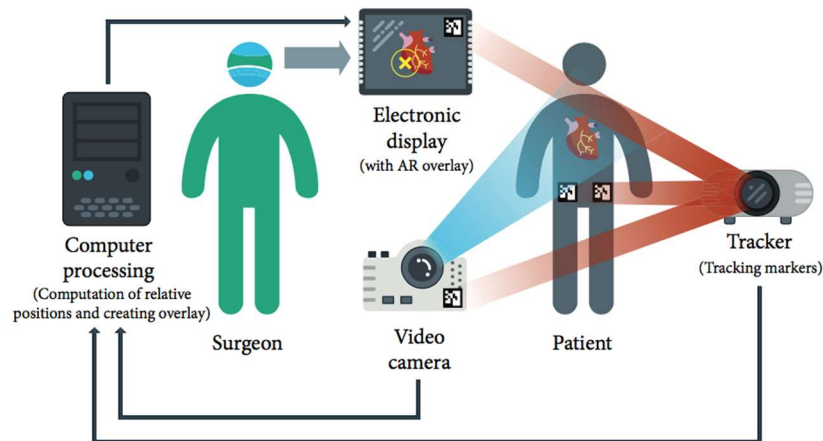
Kuva 2: Schnabelin [2007] jatkumo.

Azuma [1997] määrittelee lisätyn todellisuuden seuraavin kriteerein:

1. Yhdistää sekä todellista että virtuaalista ympäristöä
2. Interaktiivinen reaaliaikaisesti
3. Kohdistaa sekä virtuaaliset että todelliset objektit toistensa kanssa.

Azuma korostaa määrittelyssään, että lisätty todellisuus ei käsitteenä rajoita esimerkiksi näyttölaitetta vain päähän puettaviin näyttöihin, ja aisteille syötettävä data ei rajoitu pelkästään näköaistin kokemuksiin, vaan lisättyyn todellisuuteen voidaan liittää myös hajuja, makuja sekä ääniä.

Lisätty todellisuus vaatii yksinkertaisimmillaan toimiakseen näytön, seurantalaitteen, kameran ja tietokoneen. Nämä kaikki laitteet voivat myös olla samassa, kuten esimerkiksi älypuhelimessa.



Kuva 3: Perusasetelma lisätyn todellisuuden sovelluksen toteuttamisessa [Vávra et al. 2017].

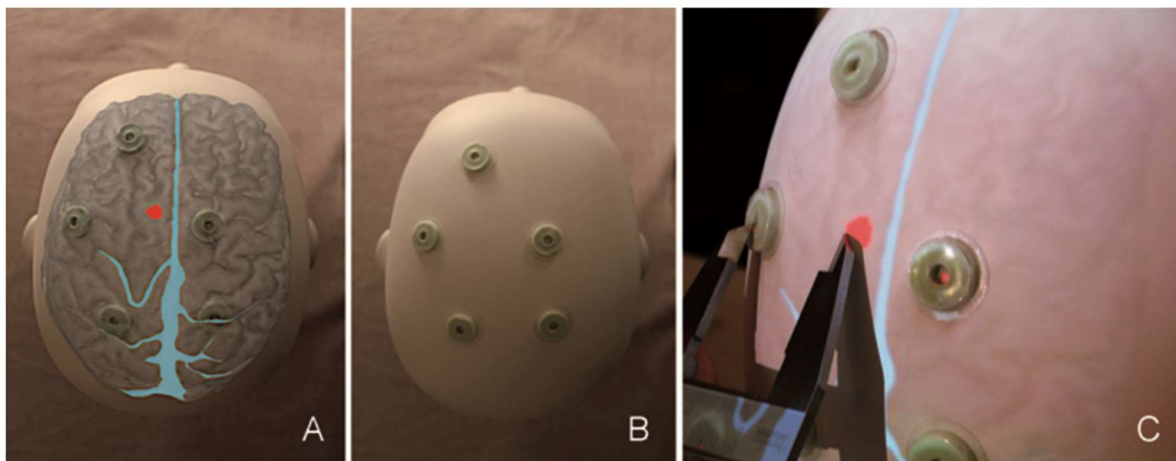
Lisätyn todellisuuden sovelluksissa on käytetty useita erilaisia näyttöjä, kuten päähän puettavia, kannettavia sekä staattisia näyttöjä. Erityisesti älypuhelin yleistymisen sekä niiden prosessointitehon kasvu ovat tehneet lisätyn todellisuuden sovelluksista yhä yleisempiä. Meolan ja muiden [2016] mukaan

näyttölaitteet olivat tyypillisesti staattisia näyttöjä sekä leikkauksessa käytettäviä mikroskooppeja, mutta joitakin esimerkkejä löytyy myös tablettitietokoneiden sekä puettavien näyttöjen käytöstä. Myös projektorina on käytetty näyttölaitteena leikkauksessa, jossa lisätyn todellisuuden hyödyntämistä tutkittiin [Besharati Tabrizi and Mahvash 2015]. Jo nykyisellään neurokirurgisissa toimenpiteissä käytetään apuna staattisia näyttöjä sekä mikroskooppeja, joten on melko luonnollista, että jo olemassa olevien laitteiden ominaisuuksia laajennetaan lisätyn todellisuuden sovelluksilla.

Jotta lisätyn todellisuuden tietokonegrafiikka reagoisi interaktiivisesti esimerkiksi käyttäjän liikkeeseen, vaativat sovellukset toimiakseen seurantalaitteita (engl. tracker) [Vávra *et al.* 2017]. Seurantalaitteet ovat kirurgisissa toimenpiteissä olleet tyypillisesti joko kameroita tai mikroskooppeja, ja ne käyttävät apunaan myös eri teknologioiden yhdistelmiä, kuten GPS-paikantimia ja gyro-skooppeja.

Grafiikan kohdistamisessa käytetään useimmiten apuna seurantamerkkejä (engl. tracking markers), joita kamera tai mikroskooppi käyttää kiintopisteiden tavoin apuna grafiikan kohdistamisessa potilaaseen nähden. Seurantamerkkeinä on käytetty esimerkiksi infrapunasäteillä potilaaseen osoitettuja pisteitä [Kawamata *et al.* 2002] sekä staattisia, heijastavia vertailumerkkejä (engl. fiducial markers) (ks. kuva 4) [Inoue *et al.* 2013; Tabrizi and Mahvash 2015]. Jotkut menetelmät eivät käytä erillisiä seurantamerkkejä lainkaan, vaan grafiikan kohdistus tehdään esimerkiksi käyttämällä potilaan kasvojen ihoa viitteenä [Meola *et al.* 2016]. Vávran ja muiden [2017] mukaan teknologian kehitys tulevaisuudessa voi kuitenkin jättää seurantamerkit tarpeettomiksi, kun seurantalaitteen keräämää dataa potilaasta voidaan prosessoida tehokkaammin ja nopeammin, ja kohdistaa grafiikka sen perusteella potilaaseen nähden. Tämän hetken teknologialla seurantamerkkien käyttäminen luo kuitenkin luotettavimman sekä nopeimman kohdistamisen [Meola *et al.* 2016].

Sekä seurantalaitteilla kerättyä että sovellukselle jo etukäteen syötettyä dataa prosessoidaan tietokoneella, joka projisoi näytölle grafiikan oikeassa koossa sekä asennossa potilaaseen nähden. Tietokoneen tulee olla langattomasti yhdistettynä muuhun laitteistoon, jotta kirurgin liikkeet eivät toimenpiteen aikana rajoitu.



Kuva 4: Heijastavia seurantamerkkejä [Tabrizi and Mahvash 2015].

## 5. Lisätty todellisuus neuronavigaation menetelmänä

Lisätty todellisuus sopii parhaiten sellaisiin leikkaustilanteisiin, joissa kudosten liikettä on hyvin vähän [Vávra *et al.* 2017]. Neurokirurgia on siis tässä mielessä ihanteellinen ala lisätyn todellisuuden sovelluksille.

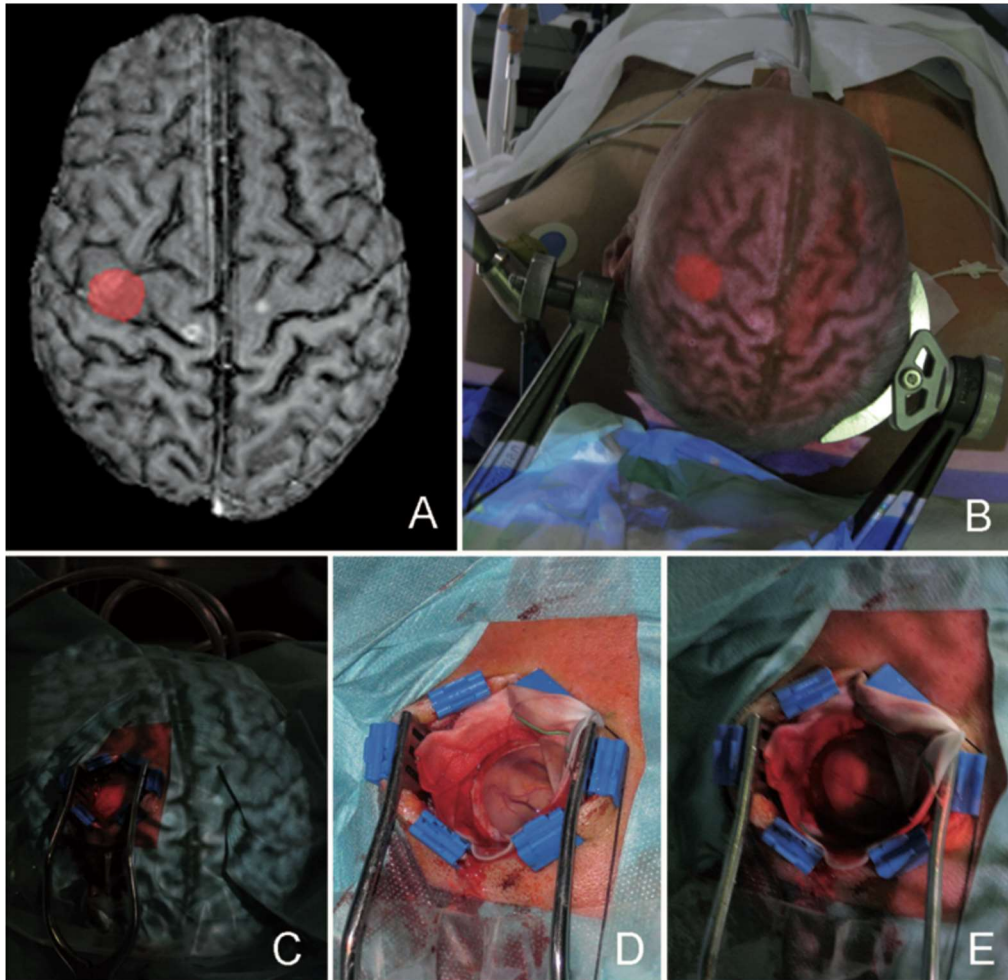
Useissa katsauksissa ja tutkimuksissa huomattiin, että lisätyn todellisuuden hyödyntäminen neurokirurgisissa toimenpiteissä toi leikkauksiin etuja [Vávra *et al.* 2015; Meola *et al.* 2016]. Suurimpina etuina mainitaan leesioiden, ja erityisesti niiden rajojen muuhun kudokseen nähden, erottamisen helpottuvan [Meola *et al.* 2016]. Tabrizi ja Mahvash [2015] korostavat erityisesti aivojen anatomian hahmottamisen helpottumista leikkaustilanteessa. Kuvan 5 kuvasarjassa havainnollistetaan, miten leesio voidaan lisätyn todellisuuden avulla projisoida potilaan aivojen päälle helposti tunnistettavasti, vaikka leesio ei erotu selkeästi paljaalla silmällä katsottaessa.

Meola ja muut [2016] mainitsevat lisätyn todellisuuden sovelluksilla olevan potentiaalia myös AVM-poikkeamaan liittyvissä leikkauksissa. Näissä leikkauksissa kirurgeille on leesioiden sijaan tärkeää erottaa potilaan verisuonet toisistaan. Lisätyn todellisuuden avulla oltiin voitu leikkaustilanteessa paikantaa luotettavasti päävaltimoita, mutta niitä ympäröivien suonien anatomiaa lisätty todellisuus ei ollut kyennyt analysoimaan ja erottelemaan. Vávra ja muut [2015] raportoivat aivojen yleisen rakenteen, kuten verisuonten, aivokudoksen poimujen sekä hermoratojen hahmottamisen helpottuneen lisätyn todellisuuden avulla.

Vávra ja muut [2015] esittävät leikkausajan lyhentyneen sekä leesioiden paikantamisen parantuneen, kun kirurgisista toimenpiteistä poistui kirurgin tarve siirtää katsettaan erillisille näytöille ja takaisin, ja toimenpiteen intuitiivisuus siten lisääntyi. Kawamata ja muut [2002] esittävät parannuksena myös sen, että kaikki kuvat voidaan lisätyn todellisuuden sovelluksissa esittää samalla näytöllä.

Lääketieteen laitekustannukset ovat suuri tekijä jatkuvasti kasvavissa terveydenhuollon kokonaiskustannuksissa. Callahanin [2008] mukaan uuden teknologian hankinta sekä jo olemassa olevien teknologioiden lisääntyvä käyttö muodostavat jopa 40-50% osuuden kustannusten vuosittaisesta kasvusta. Koska lisätyn todellisuuden sovelluksia ei vielä rutiininomaisesti käytetä neurokirurgisissa toimenpiteissä, on vaikeaa arvioida, miten niiden käyttö vaikuttaisi laitekustannuksiin [Tagaytayan *et al.* 2016]. Parhaimmassa tilanteessa modernien leikkaussalien jo olemassa olevaa laitteistoa, kuten tietokoneita, mikroskooppeja ja näyttölaitteita voitaisiin hyödyntää lisätyn todellisuuden sovelluksissa. Näin lisäkustannuksia laitteiston päivittämisen vuoksi ei syntyisi [Vávra *et al.* 2017]. Lisätyn todellisuuden sovellusten käyttöön vaadittavan peruslaitteiston kustannukset eivät kuitenkaan ole kovin korkeat. Niinpä lisätyn todellisuuden sovellusten hyödyntäminen saattaisi tulla joissakin tilanteissa jopa edullisemmaksi kuin nykyisin käytössä olevien neuronavigaation menetelmien, ja niistä voisi olla hyötyä valtioissa, joissa terveydenhuollon varallisuus on vähäistä [Lovo *et al.* 2006].

Barsomin ja muiden [2016] tekemän katsauksen mukaan lisätyn todellisuuden sovelluksista on ollut hyötyä opetustilanteissa. Katsauksen mukaan lisätyn todellisuuden sovellukset olivat nopeuttaneet prosesseja toimenpiteen simulaatiossa sekä vähentäneet simulaatiossa syntyneiden terveisiin kudoksiin kohdistuneiden vaurion määrää. Koska lisätyn todellisuuden sovellusten ominaisuuksia voidaan yhdistää jo olemassa oleviin opetusmenetelmiin, kuten harjoittelunukkeihin, on sillä suuri potentiaali antaa lisäarvoa lääketieteen opetusmenetelmiin. Leikkaustilanteiden simuloiminen kirurgian opiskelijoille on usean tutkimuksen mukaan nostanut opiskelijoiden kirurgisten taitojen tasoa sekä heidän luottamustaan niihin. Lisäksi parantuneiden taitojen on myös todistettu siirtyvän simulaatiotilanteista oikeisiin leikkaustilanteisiin [Peyre *et al.* 2006; Fernandez *et al.* 2012].



Kuva 5: Leesion havainnollistaminen tietokonegrafiikalla [Besharati Tabrizi and Mahvash 2015]

Yleisesti ottaen lisätyn todellisuuden soveltaminen neurokirurgisissa toimenpiteissä näyttää lupaavalta, ja useat tutkijat povaavatkin lisätyn todellisuuden sovellusten käytön yleistyvän neurokirurgiassa tulevaisuudessa [Kawamata *et al.* 2002; Inoue *et al.* 2013; Vávra *et al.* 2017]. Tällä hetkellä lisätyn todellisuuden sovelluksia ei kuitenkaan käytetä vielä rutiininomaisesti neurokirurgisissa toimenpiteissä, vaan niiden käyttömahdollisuuksia tutkitaan, ja sovelluksia hyödynnetään lähinnä koulutus- ja simulaatiotilanteissa [Barsom *et al.* 2016; Tagaytayan *et al.* 2016].

Eräs suurimmista ongelmista lisätyn todellisuuden sovellusten leviämisessä rutiininomaiseen käyttöön on niiden vaihtelevat latenssiajat. Toisinaan pitkät latenssiajat voivat johtua laitteiston tai verkon rajoitteista. Tagaytayan ja muut [2016] mainitsevat tutkimuksen, jossa näyttölaitteena oli käytetty tablettitietokonetta. Tutkimuksessa huomattiin tablettitietokoneen nopean liikuttamisen vaikuttavan latenssiajan kasvamiseen nostavasti. Myös langattoman verkon rajoitteet voivat kasvattaa latenssiaikaa. Latenssiaikaan liittyvien ongelmien

arvellaan kuitenkin ratkeavan teknologian kehittymisen myötä [Marcus *et al.* 2015; Tagaytayan *et al.* 2016; Vávra *et al.* 2017].

Besharati Tabrizi ja Mahvash [2015] nostavat esiin yhtenä ongelmana myös potilaan kohdistamisen järjestelmän kanssa niin, että ennen toimenpidettä hankittu data sekä potilas itse ovat linjassa toistensa kanssa. Tällä hetkellä prosessi tehdään manuaalisesti, ja sen riskeinä ovat mm. kuvan vääristyminen tai väärin kohdistuminen potilaaseen nähden. Teknologian kehittyessä potilaan kohdistaminen grafiikkaan nähden voitaneen kuitenkin automatisoida.

Marcus ja muut [2015] mainitsevat yhdeksi kriittisimmistä haasteista lisätyn todellisuuden sovellusten käyttöön otossa niin sanotun tarkkaamattoman sokeuden (engl. inattentional blindness), joka syntyy, kun leikkausalueelle ilmestyy jokin odottamaton objekti tai muutos kesken toimenpiteen. Tässä tilanteessa on mahdollista, että kirurgi ei huomaa kyseistä objektiä tietokonegrafiikan alta, ja se puolestaan voi johtaa vakaviinkin virhearviointeihin toimenpiteessä. On siis tärkeää, että sovelluksissa on jonkinlainen keino tunnistaa edellämainittuja tilanteita ja kiinnittää kirurgin huomio niihin. Koneoppimisen hyödyntäminen voisi olla yksi keinoista parantaa potilasturvallisuutta tällä saralla. Sitä voitaisiin hyödyntää esimerkiksi analysoimalla seurantalaitteen keräämää dataa näkökentästä ja tunnistamalla sen alueella esiintyviä muutoksia sekä sinne ilmestyviä objekteja.

Huolimatta siitä, että lisätyn todellisuuden sovellukset pääosin sopivat neurokirurgisiin toimenpiteisiin hyvin kudosten vähäisen liikkumisen vuoksi, voi näissäkin kudoksissa esiintyä liikettä toimenpiteiden aikana esimerkiksi painovoiman vuoksi [Tagaytayan *et al.* 2016]. Kudosten liike voi vääristää grafiikan oikeellisuutta ja siten myös kirurgin tarkkuutta operaation aikana.

Lisätyn todellisuuden sovellusten käyttöön neurokirurgisissa toimenpiteissä liittyy paljon datan käsittelyä. Vávra ja muut [2017] mainitsevatkin yhtenä sovellusten käytön haasteena suuren datamäärän esittämisen, sillä dataa on niin paljon, että kaiken sen esittäminen näytöllä voi aiheuttaa toimenpidettä suorittavalle kirurgille tilanteen, jossa ärsykyttä on liikaa. Yhtenä alan kehityskohtana voikin näin ollen pitää käytettävyyssuunnittelua, jotta käyttöliittymistä saadaan selkeitä, ja niissä esitetään vain toimenpiteen kannalta tärkeä data.

Vaikka lisätyn todellisuuden sovelluksilla nähdään olevan potentiaalia laskea neurokirurgisten toimenpiteiden laitteistokustannuksia pitkällä tähtäimellä, vaativat sovellukset ainakin näinä päivinä tietokoneita, joilla on suuri laskenta-teho, ja joiden kustannukset voivat nousta korkeiksi. Tietokoneilta kuitenkin vaaditaan, että ne voivat prosessoida tehokkaasti samanaikaisesti sekä ennen leikkausta kerättyä dataa että soveltaa sitä seurantalaitteen keräämään reaaliaikaiseen dataan mm. potilaan ja kirurgin sijainnista sekä asennosta. On kuitenkin

kin mahdollista, että lopullisissa ratkaisuissa käytetään hyväksi jo olemassa olevaa leikkaussalien varustusta. [Meola *et al.* 2016.]

Osassa sovelluksista näyttölaitteena on käytetty useimmiten päähän puettavia näyttölaitteita. Vávra ja muut [2017] nostavat esiin niihin liittyviä ongelmia, sillä puettavat laitteet voivat painaa paljon ja sen myötä aiheuttaa häiriötä opeeraatiota suorittavalle kirurgille. Laitteet voivat myös tuottaa huomattavan määrän lämpöä, ja sen myötä aiheuttaa epämukavuutta kirurgille. On erittäin todennäköistä, että puettavien näyttölaitteiden teknologinen kehitys tekee jo lähitulevaisuudessa laitteista kevyempiä sekä moninkertaistaa niiden prosessointitehoa, jolloin laitteet eivät tuota niin paljon lämpöä.

Huomionarvoista on, että keinotodellisuuden puettavien laitteiden yhteydessä joillekin käyttäjille voi tulla niin kutsutun simulaatiosairauden (engl. simulator sickness) oireita, kuten pahoinvointia, päänsärkyä ja huimausta. Simulaatiosairauden syitä ei täysin tunneta, mutta niiden arvellaan liittyvän aivojen havaitsemiin eroihin visuaalisissa ärsykkeissä. [Vávra *et al.* 2017.]

## 6. Yhteenveto

Neurokirurgisiin toimenpiteisiin liittyy nykyisellään paljon ongelmia, joita lisätyn todellisuuden sovelluksilla voidaan helpottaa. Lisätyn todellisuuden sovelluksilla on potentiaalia parantaa sekä leesioiden että niiden rajojen paikannusta terveeseen kudokseen nähden. Tietokonegrafiikalla voidaan myös helpottaa aivojen anatomian hahmottamista. Edellä mainitut parannukset lyhentävät leikkaukseen käytettyä aikaa, helpottavat leikkausalueen hahmottamista kirurgille, ja siten parantavat potilasturvallisuutta.

Ennen kuin lisätyn todellisuuden sovelluksia voidaan kuitenkaan alkaa rutiininomaisesti käyttää neurokirurgisissa toimenpiteissä, tulee useita ongelmia vielä ratkaista. Ongelmat liittyvät etenkin latenssiaikoihin sekä tilanteisiin, joissa leikkausalueella ilmenee muutoksia tai sinne ilmestyy odottamattomia objekteja. Ensimmäisenä mainitut ongelmat liittyvät laitteiden sekä niiden käyttämän verkon rajoituksiin, ja tulevat todennäköisesti ratkaisemaan itsensä teknologian kehittyessä. Sen sijaan jälkimmäisiä ongelmia voidaan selvittää esimerkiksi koneoppimista hyödyntämällä. Myös käyttöliittymäkehitys on avainasemassa, kun sovelluksia kehitetään mahdollisimman selkeiksi ja käyttäjäystävällisiksi.

Lisätyn todellisuuden sovelluksia hyödynnetään jo nykyisellään tutkimus- ja opetuskäytössä, ja tällainen käyttö tulee todennäköisesti kasvamaan tulevaisuudessa. Lisätyn todellisuuden avulla toteutetut simulaatiot ovat verrattain edullisia ja helppoja toteuttaa, ja teknologiaa voidaan hyödyntää jo nykyisin käytössä olevien opetusmenetelmien lisänä.

Jo nyt on kuitenkin lupaavia tuloksia erilaisista tutkimuksista, joissa leesioita on voitu poistaa onnistuneesti potilaiden aivoista ilman komplikaatioita

lisätyn todellisuuden sovellusten avulla. On siis hyvin todennäköistä, että lisätyn todellisuuden sovelluksia tullaan jo lähitulevaisuudessa näkemään neurokirurgiassa enemmän, ja ne parantavat potilasturvallisuutta entisestään.

## Viiteluettelo

- Ronald T. Azuma. 1997. A survey of augmented reality. In: *Proc. of the Presence: Teleoperators and Virtual Environments* 6, 4, 355–385.
- Esther Z. Barsom, Maurits Graafland and Marlies P. Schijven. 2016. Systematic review on the effectiveness of augmented reality applications in medical training. *Surgical Endoscopy* 30, 4174–4183.
- Leila Besharati Tabrizi and Mehran Mahvash. 2015. Augmented reality-guided neurosurgery: accuracy and intraoperative application of an image projection technique. *Journal of Neurosurgery* 123, 206–211.
- Daniel Callahan. 2008. Health Care Costs and Medical Technology, In: Mary Crowley (ed.) *From Birth to Death and Bench to Clinic: The Hastings Center Bioethics Briefing Book for Journalists, Policymakers, and Campaigns*, Garrison, NY: The Hastings Center, 79–82.
- Walter E. Dandy. 1918. Ventriculography following the injection of air into the cerebral ventricles. *Annals of Surgery* 68, 5–11.
- Gladys L. Fernandez, David W. Page, Nicholas P. Coe, Patrick C. Lee, Lisa A. Patterson, Loki Skylizard, Myron St. Louis, Marisa H. Amaral, Richard B. Wait, and Neal E. Seymour. 2012. Boot camp: educational outcomes after 4 successive years of preparatory simulation based training at onset of internship. *Journal of Surgical Education* 69, 2, 242–248.
- Jens Haase. 1999. Neuronavigation. *Child's Nervous System* 15, 755–757.
- Daisuke Inoue, B. Cho, M. Mori, Yuichiro Kikkawa, Toshiyuki Amano, Akira Nakamizo, Koji Yoshimoto, Masahiro Mizoguchi, Morimasa Tomikawa, Jaesung Hong, Makoto Hashizume and Tomio Sasaki. 2013. Preliminary study on the clinical application of augmented reality neuronavigation. *Journal of Neurological Surgery* 74, 71–76.
- Marcel Ivanov and Alexandru Vlad Ciurea. 2009. Neuronavigation. Principles. Surgical technique. *Journal of Medicine and Life* 2, 1, 29–35.
- Takakazu Kawamata, Hiroshi Iseki, Takao Shibasaki and Tomokatsu Hori. 2002. Endoscopic augmented reality navigation system for endonasal transsphenoidal surgery to treat pituitary tumors: technical note. *Neurosurgery* 50, 6, 1393–1397.
- Eduardo E. Lovo, Juan C. Quintana, Manuel C. Puebla, Gonzalo Torrealba, José L. Santos, Isidro H. Lira and Patricio Tagle. 2006. A novel, inexpensive



- method of image coregistration for applications in image-guided surgery using augmented reality. *Operative Neurosurgery* 2, 60, 366–372.
- Hani J. Marcus, Philip Pratt, Archie Hughes-Hallett, Thomas P. Cundy, Adam P. Marcus, Guang-Zhong Yang, Ara Darzi and Dipankar Nandi. 2015. Comparative effectiveness and safety of image guidance systems in neurosurgery: a preclinical randomized study. *Journal of Neurosurgery* 123. 307–313.
- Antonio Meola, Fabrizio Cutolo, Marina Carbone, Federico Cagnazzo, Mauro Ferrari and Vincenzo Ferrari. 2017. Augmented reality in neurosurgery: a systematic review. *Neurosurgical Review* 40, 537–548.
- Paul Milgram and Fumio Kishino. 1994. A taxonomy of mixed reality visual displays. *IEICE Transactions on Information Systems*, E77-D, 12, 1–15.
- Pertti Mustajoki ja Jarmo Kaukua. 2003. *Senkka ja 100 muuta tutkimusta*. Kustannus Oy Duodecim.
- Abhilash Pandya, Mohammad-Reza Siadat and Greg Auner. 2005. Design, implementation and accuracy of a prototype for medical augmented reality. *Computer Aided Surgery* 10, 1, 23–35.
- Sarah E. Peyre, Christian G. Peyre, Maura E. Sullivan and Shirin Towfigh. 2006. A surgical skills elective can improve student confidence prior to internship. *Journal of Surgical Research* 133, 11–15.
- Mark Aurel Schnabel, Xiangyu Wang, Hartmut Seichter and Tom Kvan. 2007. From virtuality to reality and back. In: *Proc. of the International Association of Societies of Design Research*, 1–15.
- Raniel Tagaytayan, Arpad Kelemen and Cecilia Sik-Lanyi. 2016. Augmented reality in neurosurgery. *Archives of Medical Science* 1–7.
- P. Vaivara, J. Roman, P. Zonca, Peter Ihnat, M. Nemec, Jayant Kumar, Nagy Habib, and Ahmed El-Gendi. 2017. Recent development of augmented reality in surgery: a review. *Journal of Healthcare Engineering* 2017, Article ID 4574172, 19.
- Duodecim terveyskirjasto: Lääketieteen sanasto. 2017. Kustannus Oy Duodecim. [http://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p\\_artikkeli=ltt01891](http://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p_artikkeli=ltt01891) (Luettu 15.11.2017)

# Katsaus roskapostisuodattimiin

**Mikko Helin**

## Tiivistelmä.

Sähköposti on nopeutensa ja halpuutensa ansiosta vakiintunut ympäri maailmaa tärkeiden asiakirjojen ja henkilökohtaisten tietojen välittämiseen, mutta helpouden varjopuolena on sähköpostin välityksellä liikkuva roskaposti, joka aiheuttaa yrityksille ja yksityishenkilöille ajanhukkaa ja joskus vakaviakin tietoturvauhkia. Ongelmaa ratkaisemaan on kehitetty roskapostisuodattimia, jotka tunnistavat ja siirtävät roskapostit omaan kansioonsa käyttäjän puolesta. Roskapostisuodattimia on monia, ja jokaisella niistä on perustanaan hyvin erilaisia toimintaperiaatteita, vahvuuksia ja heikkouksia.

Vaikka roskapostisuodattimet ovat nykyisin hyvin tarkkoja, on suodattimissa yhä ratkaisemattomia ongelmia, jotka voivat pahimmillaan pilata koko suodattimen toiminnan. Tässä tutkielmassa tarkastellaan kahta yleistä suodatus-tekniikkaa (naiivi Bayes-luokittelija ja tukivektorikone (SVM)) ja vertaillaan niiden vahvuuksia ja heikkouksia, ja tarkastellaan kirjallisuuden perusteella, mitä ratkaisemattomia ongelmia niissä vielä on, ja mitä seurauksia ongelmilla on. Tukivektorikone osoittautui suosionsa ja tarkkuutensa perusteella parhaana pidetyksi menetelmäksi, mutta naiivi Bayes-luokittelija taas oli luotettavampi suurien datamäärien luokittelussa sekä vaatii vähemmän tehoa. Väärien positiivisten luokitusten välttäminen osoittautui kirjallisuuskatsauksen perusteella yleisesti tärkeäksi tutkimusaiheeksi roskapostisuodatuksessa.

**Avainsanat ja -sanonnat:** sähköpostisuodatin, koneoppiminen, spam, asiakirjojen luokittelu, luokittelualgoritmi, naiivi Bayes, tukivektorikone.

## 1. Johdanto

Sähköposti on vakiintunut olennaiseksi osaksi yksityishenkilöiden, yritysten ja julkisten organisaatioiden viestintää. Digitalisoitumisen ja internetin kasvavan käyttäjämäärän vuoksi yhä useampi tärkeä asiakirja välitetään juuri sähköpostin avulla, sillä se on helppo, halpa ja nopea väline, ja se on lähes korvannut paperipostin. Käyttäjää kuitenkin häiritsevät kansiota täyttävät, ei-halutut viestit kuten mainokset, huijausviestit ja muu roskaposti eli *spam*.

Roskaposti sisältää usein käyttäjän kannalta ei-toivottuja mainoksia, mutta joskus myös tietojenkalasteluyrityksiä tai haittaohjelmia, joista voi olla suurta vahinkoa viestin vastaanottajalle.

Sisällöltään harmittomienkin roskapostiviestien läpi kahlaaminen voi hukata aikaa ja rahaa yksityisiltä henkilöiltä ja organisaatioilta. Ongelma ei ole pieni: maailmanlaajuisesti roskapostin synnyttämä häiriö oli aiheuttanut jo vuoteen 2004 mennessä jopa 50 miljardin USA:n dollarin kulut [Almeida 2011]. Hoodan ja muiden [2015] mukaan päivittäin lähetetään 12,4 miljardia roskapostiviestiä, ja vuodessa vastaanotetaan 2200 roskapostiviestiä henkilöä kohti, ja vuosittain tuhannen työntekijän kokoinen yritys vastaanottaa 2,1 miljoonaa roskapostiviestiä.

Tätä ongelmaa ratkaisemaan on kehitetty sähköpostisuodattimia, jotka tunnistavat automaattisesti roskapostin ja sijoittavat sen suoraan roskapostikan-sioon, jolloin käyttäjä voi keskittyä vain tärkeisiin viesteihin eli *ham*-viesteihin. Vaikka suodatinohjelma ei ymmärrä viestin sisältöä, osaavat nykyiset suodattimet erottavat spam- ja ham-viestit oikein jopa yli 99% tarkkuudella [Daeeff *et al.* 2016].

Tässä tutkielmassa tehdään katsaus nykyisiin sähköpostisuodattimiin ja selvitetään, mitkä suodatinmenetelmät ovat tällä hetkellä tehokkaimmat ja yleisimmät, tarkastellaan näiden suodattimien toimintaperiaatteita ja vertaillaan suodattimien eroja ja erojen syitä.

Aihe on rajattu roskapostisuodattimien tutkimuksen perusteella kahteen merkittävänä pidettyyn roskapostisuodattimeen: *tukivektorikone* ja *naiivi Bayes-luokittelija*.

Kirjallisuuskatsauksen mukaan tukivektorikonetta pidetään parhaana ja tarkkuutensa ja suosionsa perusteella, mutta silti muille menetelmille, kuten naiiville Bayes-luokittelijalle on olemassa omat sovelluksensa, sillä sopivan suodatinmenetelmän valinta riippuu ratkaistavasta ongelmasta ja datasta, johon suodatinta sovelletaan: tarkkuudestaan huolimatta tukivektorikone vaatii paljon tehoja, on heikko suuren datamäärän luokittelussa ja suuri määrä poikkeuksia datassa voi sekoittaa sen toiminnan, kun taas naiivi Bayes-luokittelijalle selviää edellä mainituista tilanteista, mutta ei ole yhtä tarkka ja voi olla altis bayesilaiselle myrkytykselle.

Tärkeäksi tulevaisuuden tutkimuskohteeksi osoittautui väärin positiivisten luokitusten välttäminen, sillä hyödyllisten viestien hävittäminen on käytännön kannalta potentiaalisesti kaikkein pahimmat seuraukset – jopa pahemmat kuin muutaman roskapostin pääsemisellä suodattimen läpi silloin tällöin, vaikka roskapostiviestit saattavatkin sisältää vakavia tietoturvariskejä ja häiritsevät käyttäjän toimintaa.

Luvussa 2 esitellään aikaisempi tutkimus ja perustellaan tarkemmin tarkasteltujen suodatinmenetelmien, tukivektorikoneen ja naiivin Bayes-luokittelijan valintaa tutkielman aiheeksi.

Luvussa 3 esitellään roskapostisuodattimien yleinen toiminta ja olennaisia termejä. Luvussa 4 esitellään yleiset arviointikriteerit, joilla roskapostisuodattimien toimivuutta voidaan vertailla ja mitata. Luvussa 5 esitellään tukivektori-kone ja naiivi Bayes-luokittelija pääpiirteittäin sekä esitellään kummankin heikkouksia ja vahvuuksia. Luvussa 6 vertaillaan näitä kahta menetelmää keskenään ja luvussa 7 pohditaan, mitä erot tarkoittavat käyttäjän ja tulevaisuuden tutkimuksen kannalta, ja mitä yleisiä tutkittavia ongelmia havaittiin.

## 2. Aikaisempi tutkimus

Roskapostisuodatus on eräs koneoppimisen sovellusalue. Mohrin ja muiden [2012] mukaan koneoppimisen voi määritellä suurpiirteisesti tarkoittamaan laskennallisia menetelmiä, joissa käytetään kokemusta parantamaan suorituskkyä tai tekemään tarkkoja ennusteita.

*Kokemuksella* tarkoitetaan tässä tapauksessa menneisyyden tietoa, joka on saatavilla oppijalle elektronisen datan muodossa ja joka on kerätty ja annettu analysoitavaksi. Tämä data voi olla ihmisen leimaaman opetusdatan muodossa (training set, learning data), tai muun tyyppisen tiedon muodossa, joka on saatu interaktiossa ympäristön kanssa. Kaikissa tapauksissa datan laatu ja koko ovat kriittisiä oppijan tekemien ennusteiden onnistumiselle. *Esimerkit* (example) tarkoittavat roskapostisuodatuksen ollessa kyseessä käsiteltäviä sähköpostiviestejä, ja *leimat* (label) viestin oikeaa kategoriaa, jotka ovat binäärisessä roskapostisuodatuksessa nimellä spam ja ham.

Mujtaba ja muut [2017] tutkivat vuosien 2006-2016 välillä julkaistuja tutkimuksia roskapostisuodattimista, ja selvittivät tutkimustrendien perusteella yleisimmät suodatintyypit sekä niiden arvioimiseen käytetyt kriteerit. Kyseisen tutkimuksen perusteella tähän tutkielmaan on valittu suodattimista kahden menetelmän kokoinen nykyisiä tutkimustrendejä edustava otos.

Sähköpostiluokittelutekniikoissa sovelletaan ainakin seuraavia viittä koneoppimisen tyyppiä:

1. *Ohjattu koneoppiminen* (supervised machine learning): oppija vastaanottaa leimattuja esimerkkejä opetusdataksi ja tekee ennustuksia kaikille datapisteille, joita ei ole vielä nähnyt. Roskapostisuodatuksessa se tarkoittaa sitä, että oppija saa roskaposteiksi ja ei-roskaposteiksi valmiiksi merkittyä opetusdataa, jonka perusteella se oppii ennustamaan tulevien sähköpostiviestien kohdalla, mihin luokkaan ne kuuluvat [Mohri *et al.* 2012].

2. *Semiohjattu koneoppiminen* (semi-supervised machine learning): oppija vastaanottaa opetusdataa, joka koostuu sekä leimatusta että leimaamattomasta datasta, ja tekee ennusteita ja tekee sen perusteella ennusteita toistaiseksi havaitsemattoman datan varalle [Mohri *et al.* 2012].
3. *Ohjaamaton koneoppiminen* (unsupervised machine learning): oppija saa vain leimaamatonta dataa ja tekee sen perusteella ennusteita toistaiseksi havaitsemattoman datan varalle [Mohri *et al.* 2012].
4. *Sisältöpohjainen koneoppiminen* (content-based machine learning): avainsanoja (keyword) sähköposteissa käytetään luokitteluun [Mujtaba *et al.* 2017].
5. *Tilastollinen oppiminen* (statistical learning): jokaiselle avainsanalle annetaan todennäköisyys tai jokin pisteytys, ja kokonaispistemäärää tai todennäköisyyttä käytetään luokittelussa [Mujtaba *et al.* 2017].

Näistä yleisimmin roskapostisuodattamiseen käytetty tyyppi on ohjattu koneoppiminen [Mujtaba *et al.* 2017]. Tämä tutkielma on rajattu kyseiseen tyyppiin, ja siksi siihen perustuvat myös tässä tutkielmassa käsitellyt suodatintekniikat.

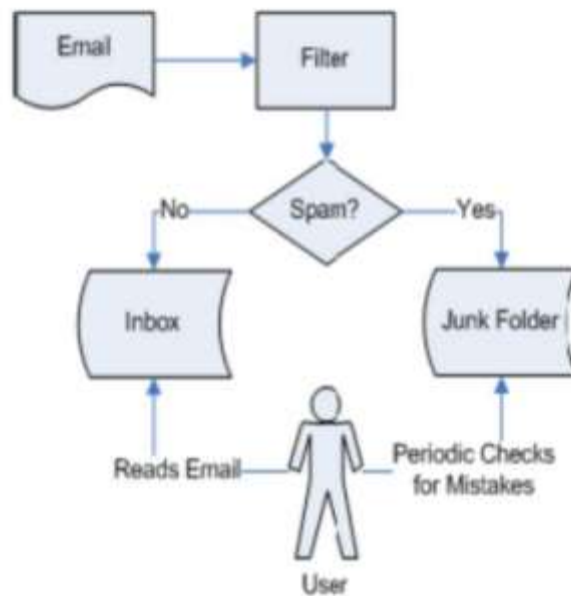
Tässä tutkielmassa ei erotella tietojenkalastusviestejä ja muun tyyppisiä roskaposteja toisistaan, vaikka jotkin tutkimukset keskittyvät niihin erikseen, vaan kaikenlaiset huijausviestit, mainosviestit ja muut käyttäjälle hyödyttömät tai haitalliset viestit käsitellään yhdessä yleistermillä roskaposti eli spam. Tutkielma keskittyy pääasiassa binääriseen luokitteluun, eikä esimerkiksi moniluokitteluun, jossa puolestaan olisi useampia kategorioita kuin spam ja ham.

Mujtaban ja muiden [2017] mukaan yleisin ja myös tehokkain roskapostisuodatinmenetelmä on tukivektorikone eli SVM, seuraavana *päättöspuu* ja kolmantena naiivi Bayes-luokittelija.

Vain tukivektorikone ja naiivi Bayes-luokittelija on kuitenkin valittu vertailtavaksi tutkielman tiivistämiseksi. Tukivektorikone ja naiivi Bayes-luokittelija ovat kummatkin kolmen tehokkaimman menetelmän joukossa ja ovat myös sopivan erilaisia vahvuuksiltaan ja heikkouksiltaan. Suosittuina menetelminä ne edustavat tämän hetken tutkijoita kiinnostavia aiheita ja tulevaisuuden tutkimussuuntia.

### 3. Roskapostisuodattimen toiminta

Roskapostisuodatus on sähköpostin järjestelyä määriteltyjen kriteerien mukaisesti. Suodatinohjelma saa syötteen sähköposteja, ja se joko päästää ne läpi sellaisenaan käyttäjän postilaatikkoon, erottelee sähköpostit eri kansioihin, ohjaa uudelleen toiseen osoitteeseen. (Kuva 1). Jotkut suodattimet jopa muuntelevat viestin sisältöä. [Chhabra *et al.* 2010].



Kuva 1. Roskapostisuodatuksen kuvaus pääpiirteittäin [Hooda *et al.* 2015].

Suodatin ei ymmärrä sähköpostiviestien sisältöä, mutta se voidaan opettaa luokittelemaan roskapostit spam-luokkaan ja tarpeelliset viestit ham-luokkaan löytämällä niistä säännönmukaisuuksia, jotka erottavat viestit kahteen eri luokkaan. Jotta suodatin tietää, minkälaisia roskapostiviestit ja hyödylliset viestit ovat, suodatinohjelmalle syötetään valmiiksi valikoitua opetusdataa, jonka avulla se oppii tunnistamaan spam- ja ham-viestit. Sahin ja Parmarin [2017] mukaan suositeltava roskapostin tunnistamisen arkkitehtuuri sisältää seuraavat vaiheet: datan kerääminen, datan prosessointi, piirteiden erottaminen ja datan luokittelutulosten analysointi.

Suodatin käsittelee opetusdatan, ja opettelee tunnistamaan piirteitä, joita valmiiksi merkityissä spam- ja ham-luokkiin erotelluissa viesteissä on. Piirteitä kerätään aina koko viestistä, otsikosta ja aiheesta lähtien.

*Piirteet* (features) tarkoittavat koneoppimisessa joukkoa attribuutteja, usein esitettynä vektorina, joka liittyy tässä tapauksessa sähköpostiviesteihin. Sähköpostiviestin ollessa kyseessä joitakin relevantteja piirteitä voivat olla esimerkiksi viestin pituus, lähettäjän nimi, otsikon joitakin ominaisuuksia, joidenkin sanojen esiintyminen koko viestin sisällössä tai muuta vastaavaa. [Mohrin *et al.* 2012].

Piirteiden erottamisessa kuitenkin tulisi minimoida piirteiden määrä ja välttää turhien piirteiden luomista, jotta luokittelun tarkkuus ei kärsi [Temitayo *et al.* 2012].



Kuva 2. Sähköpostiluokittelun yleinen arkkitehtuuri [Mujtaba *et al.* 2017].

Mujtaban ja muiden [2017] mukaan sähköpostiluokitteluprosessi jakautuu kolmeen selvästi erotettavaan tasoon:

- esikäsittely (pre-processing)
- oppiminen (learning)
- luokittelu (classification).

Kuvassa 2 on esitetty sähköpostiluokittelun yleinen arkkitehtuuri. Kuvatuissa vaiheissa tapahtuu seuraavaa:

1. Datakokoelman (data collection) kerääminen, jossa on sekä spam- että ham-viestejä.
2. Alkionimittäminen (tokenization): datajoukko "puhdistetaan" esikäsittelyssä: sähköpostin sisältö muunnetaan alkionimiksi. Tarpeettomat sanat poistetaan käsiteltävän datan minimoimiseksi.

3. Stemmaus (stemming) ja perusmuotoistaminen (lemmatization) sovelletaan alkionimiin muuntaen ne perusmuotoihinsa.
4. Opetustasolla piirrejoukot muodostetaan ja piirteet erotetaan. Sähköpostiluokittelussa hyvä piirteiden erottaminen on olennaista oppimisprosessin tarkkuuden ja tehokkuuden kannalta.
5. Piirteet, jotka parhaiten erottelevat luokat toisistaan, valitaan tehokkuuden ja luokittelun tarkkuuden parantamiseksi.
6. Luokittelutasolla rakennettu luokittelija luokittelee saapuvaa sähköpostia tiettyyn luokkaan, kuten spam tai ham.

Suodattimilla suoritettavia kokeita tehdään usein valmiilla, internetissä saatavilla olevilla roskapostitietokannoilla, kuten Spambase, SpamAssassin ja TREC.

Valmiit tietokannat ovat toisinaan kuitenkin useita vuosia vanhoja, ja jotkin menetelmät vaativat ajan tasalla olevaa dataa; yhtenä menetelmänä onkin myös reaaliaikainen oppiminen täysin tuoreesta datasta.

#### **4. Roskapostisuodattimien yleiset arviointikriteerit**

Kriteerit, joilla suodattimien suorituskkyä useimmiten arvioidaan, perustuvat pohjimmiltaan siihen, ovatko suodatetun viestin todellinen luokka ja suodattimen viestille ennustama luokka samat. Yleisesti luokittelun toimivuutta arvioidaan sen tarkkuuden ja nopeuden perusteella [Chhabra *et al.* 2010].

Tarkintaan suodatin ei luokittele aivan jokaista viestiä oikeaan luokkaansa. Arvioidessa roskapostisuodatinta otetaan huomioon sekä oikeat että väärät luokitukset ja niiden todennäköisyys. Suodatin on sitä tarkempi mitä harvemmin se erehtyy luulemaan roskapostia hyödylliseksi tai hyödyllistä viestiä roskapostiksi.

Oikeiden positiivisten luokitusten määrä (true positive) on korkea silloin, kun suodattimen ennustama luokittelu ja viestien todellinen luokka ovat kummatkin luokkaa "spam", eli suodatin on tunnistanut roskapostin roskapostiksi. Vastaavasti väärin positiivisten luokitusten määrä (false positive) tarkoittaa ham-luokkaan luokiteltujen spam-viestien määrää, eli suodatin on virheellisesti luokitellut roskapostin hyödylliseksi viestiksi.

Oikea negatiivinen luokitus (true negative) ja väärä negatiivinen (false negative) puolestaan tarkoittavat oikein ennustettujen ham-viestien määrää, eli hyödyllisen viestin luokittelemista hyödylliseksi, ja väärin ennustettujen ham-viestien määrää, eli hyödyllisen viestin luokittelemista roskapostiksi. Havainnollistukseksi edellä selitetyt termit ovat piirrettynä kuvassa 3 esitettyyn sekaannusmatriisiin.



		Sähköpostin todellinen luokka	
		Spam	Ham
Sähköpostin ennustettu luokka	Spam	Oikea positiivinen	Väärä positiivinen
	Ham	Väärä negatiivinen	Oikea negatiivinen

Kuva 3. Sekaannusmatriisi.

Muita tutkimuksissa käytettyjä kriteerejä on esitelty kuvassa 4. Kun luokittelutyyppi binäärinen, kriteereitä ovat seuraavat:

- **Täsmällisyys** (precision) luokittelun yleinen tehokkuus.
- **Palautus** (recall) luokkayhteensopivuus datan leimojen ja luokittelijan antamien positiivisten leimojen kanssa.
- **F-arvo** (f-measure) luokittelijan tehokkuus positiivisten leimojen tunnistamisessa.
- **Tarkkuus** (accuracy) datan positiivisten leimojen ja luokittelijan antamien leimojen suhde.
- **Osuuus** (specificity), kuinka tehokkaasti luokittelija tunnistaa negatiiviset leimat.

Kun luokittelutyyppinä on moniluokitus, niin kriteereitä ovat seuraavat:

- **Area Under Curve (AUC)**, luokittelijan kyky välttää väärää luokitusta.
- **Keskimääräinen tarkkuus** (average accuracy), keskimääräinen luokka-kohtainen tehokkuus.
- **Kokonaisvirhemäärä** (error rate), keskimääräinen luokkakohtainen luokitteluvirhe.
- **Täsmällisyys- $\mu$**  (precision- $\mu$ ), datan luokkaleimojen ja luokittelijan antamien leimojen vastaavuus kun lasketaan tekstikohtaisten päätösten yhteenlasketuista summista.
- **Palautus- $\mu$**  (recall- $\mu$ ), luokittelijan tehokkuus tunnistaa luokkaleimoja kun lasketaan tekstikohtaisten päätösten yhteenlasketuista summista.
- **F-arvo- $\mu$**  (F-Measure- $\mu$ ), datan positiivisten leimojen ja luokittelijan antamien leimojen väliset suhteet perustuen tekstikohtaisten päätösten yhteenlasketuista summista moniluokitustapauksissa.

Edellisten kriteerien lisäksi voidaan asiayhteyden mukaan arvioida esimerkiksi suoritusaikaa [Daeef et al. 2016].

Yleisesti tutkimuksissa pääpaino on väärin positiivisten ja väärin negatiivisten luokitusten määrien vertailussa. Abu-Nimehin ja muiden [2007] mukaan sähköpostisuodatuksessa väärin positiivisiin ja väärin negatiivisiin luokituksiin kiinnitetään enemmän huomiota, kun tutkitaan luokittelijan suori- tuskyykyä, sillä väärät positiiviset luokitukset tulevat tosimaailmassa kalliim- miksi kuin väärät negatiiviset luokitukset. Kokonaisvirhemäärä ei kuitenkaan kerro kaikkea, vaan tarkemman vertailun saa aikaiseksi vertailemalla AUC- arvoa.

S. No	Measure	Classification Type	Formula	Evaluation Focus
1	Precision	Binary	$Precision = \frac{tp}{tp + fp}$	Overall effectiveness of a classifier
2	Recall	Binary	$Recall = \frac{tp}{tp + fn}$	Class agreement of the data labels with the positive labels given by the classifier
3	F-Measure	Binary	$Specificity = 1 / 2(\frac{tp}{tp + fn} + \frac{tn}{tn + fp})$ $F - Measure = \frac{(\beta 2 + 1)tp}{(\beta 2 + 1)tp + \beta 2 fn + fp}$	Effectiveness of a classifier to identify positive labels
4	Accuracy	Binary	$Accuracy = \frac{(tp + tn)}{(tp + fn + fp + tn)}$	Relations between data's positive labels and those given by a classifier
5	Specificity	Binary	$Specificity = \frac{tn}{fp + tn}$	How effectively a classifier identifies negative labels
6	AUC	Binary	$Specificity = 1 / 2(\frac{tp}{tp + fn} + \frac{tn}{tn + fp})$	Classifier's ability to avoid false classification
7	Average Accuracy	Multi-Class	$\sum_{i=1}^l (\frac{tp_i + tni}{tp_i + fni + fpi + tni}) / l$	The average per-class effectiveness of a classifier
8	Error Rate	Multi-Class	$\sum_{i=1}^l (\frac{fpi + fni}{tp_i + fni + fpi + tni}) / l$	The average per-class classification error
9	Precision <sub>μ</sub>	Multi-Class	$\frac{\sum_{i=1}^l tp_i}{\sum_{i=1}^l (tp_i + fp_i)}$	Agreement of the data class labels with those of a classifiers if calculated from sums of per-text decisions
10	Recall <sub>μ</sub>	Multi-Class	$\frac{\sum_{i=1}^l tp_i}{\sum_{i=1}^l (tp_i + fn_i)}$	Effectiveness of a classifier to identify class labels if calculated from sums of per-text decisions
11	F-Measure <sub>μ</sub>	Multi-Class	$\frac{(\beta 2 + 1) Precision_{\mu} Recall_{\mu}}{(\beta 2 + 1) Precision_{\mu} + Recall_{\mu}}$	Relations between data's positive labels and those given by a classifier based on sums of per-text decisions

Kuva 4. Suodattimien yleiset arviointikriteerit ja niiden laskutoimitukset. TP = oikeat positiiviset, FP = väärät positiiviset, FN = väärät negatiiviset ja TN = oikeat negatiiviset luokitukset [Mujtaba *et al.* 2017].

## 5. Kaksi tehokkainta roskapostisuodatinmenetelmää

Seuraavaksi esitellään naiivi Bayes-luokittelija ja tukivektorikone ja niiden vahvuudet ja heikkoudet.

### 5.1 Naiivi Bayes-luokittelija

Naiivi Bayes-luokittelija (NB) on bayesilainen verkko sovellettuna luokittelu-tehtävään [Sahami *et al.* 1998]. NB on perinteinen ja suosittu menetelmä asiakirjojen luokitteluun. Se perustuu ehdolliseen todennäköisyyteen, ja olettaa piirteiden olevan riippumattomia toisistaan.

NB oppii opetusdatasta mitkä sanat ilmenevät useammin spam-viesteissä ja mitkä sanat useammin ham-viesteissä. Tämän jälkeen yksittäisillä sanoilla on tietty todennäköisyys esiintyä spam-viestissä  $P(Sana | Spam)$  tai ham-viestissä  $P(Sana | Ham)$ . Kaikille sanoille on siis olemassa todennäköisyystaulukko, joka perustuu sanojen ilmenemistiheyteen spam ja ham-luokissa. Myös jokaisella viestillä on yleisesti todennäköisyys kuulua luokkaan spam,  $P(Spam)$ , tai ham,  $P(Ham)$ .

NB arvioi ehdollisen todennäköisyyden avulla, onko tietty viesti roskapostia vai ei laskemalla siinä olevien sanojen todennäköisyydet kuulua roskaposti-viestiin. Mitä enemmän spam-luokkaan kuuluvia sanoja se sisältää, sitä todennäköisemmin viesti on roskaposti.

Jos viestin todennäköisyys olla spam on suurempi kuin luottamusraja  $T$ , viesti kuuluu luokkaan spam. Luottamusraja voidaan määritellä vaaditun tarkkuuden perusteella. Mitä korkeampi  $T$  on, esimerkiksi 0.5, silloin kaikkien viestin sanojen yhteenlaskettu todennäköisyys tulee olla yli 50%, että se merkitään roskapostiksi.

Tästä seuraa, että mitä korkeampi  $T$ -raja on, sitä korkeampi on väärän negatiivisen luokittelun riski, jolloin roskaposti pääsee helpommin suodattimen läpi postilaatikkoon, ja mitä matalampi  $T$ -raja on, sitä korkeampi väärän positiivisen luokittelun riski on, jolloin suodatin tuomitsee herkemmin sellaisia viestejä roskapostiksi, jotka eivät sitä välttämättä ole.

Todennäköisyys sille, että viesti kuuluu luokkaan *spam*, sillä ehdolla että siinä on tietty sana, lasketaan Bayesin teoreeman mukaisesti  $P(Spam | Sana)$ :

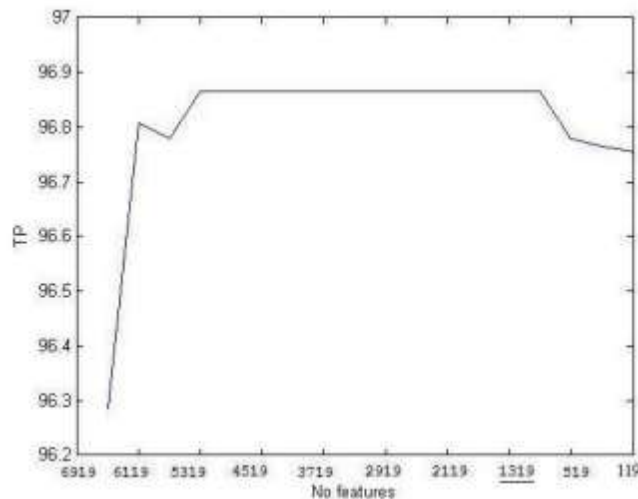
$$P(Spam | Sana) = \frac{P(Sana | Spam)P(Spam)}{P(Sana | Spam)P(Spam) + P(Sana | Ham)P(Ham)}.$$

Lisäksi nykyisissä Bayes-luokittelijaan perustuvissa suodatinohjelmissa jätetään huomiotta turhat sanat kuten "on" ja "ja", jotka eivät liity roskapostin tunnistamiseen, sekä harvinaiset sanat, jotka ilmenevät vain sattumalta toisessa luokassa.

Vahvuutena naiivi Bayes-luokittelijalla on se, että se sopii suuriin datamassoihin. NB osaa käsitellä suuriakin määriä piirteitä, joskin piirteiden liian suuri

määrä voi myös heikentää tarkkuutta, jolloin piirteitä tulee karsia. Menetelmä on myös joustava ja sopii hyvin erilaisiin luokittelutehtäviin, ja siksi sitä käytetään laajasti.

NB on myös melko tarkka: Esmaeili ja muut [2017] testasivat oikeiden luokitusten määrää suhteessa piirteiden määrään, ja tuloksena oli 96.87% oikeita positiivisia luokituksia ja 1319 piirrettä oli optimaalinen määrä tarkkuudelle (Kuva 5). [Esmaeili *et al.* 2017].



Kuva 5. Y-akselilla: oikeiden positiivisten (TP) luokitusten prosentuaalinen osuus. X-akselilla: piirteiden määrä. [Esmaeili *et al.* 2017].

NB:n eräs tunnettu ongelma on piirreavaruuden suuri dimensionaalisuus [Androutsopoulos *et al.* 2004]. NB:n suorituskky riippuu suuresti valituista piirteistä opetusvaiheessa. Luokittelun onnistumista määrää paljon luokittelijan terminvalikoimismenetelmä, joka täsmentää, miten piirteitä havaitaan datasta. Mitä parempi terminvalikoimismenetelmä, sitä tarkempi luokittelu on. [Almeida *et al.* 2011].

Suurin toistaiseksi ratkaisematta oleva ongelma on yhä bayesilainen myrkytys sekä käänteinen myrkytys. NB voidaan "myrkyttää" antamalla riittävästi ham-luokkaa muistuttavia sanoja roskapostiviestien mukana, jolloin NB ei tunnista viestiä roskapostiksi. Käänteinen myrkytys puolestaan tarkoittaa, että roskapostin lähettäjä opettaa NB:n tahallaan luulemaan ham-viestejä roskapostiksi lisäämällä spam-luokassa ilmeneviä sanoja tärkeän viestin joukkoon.

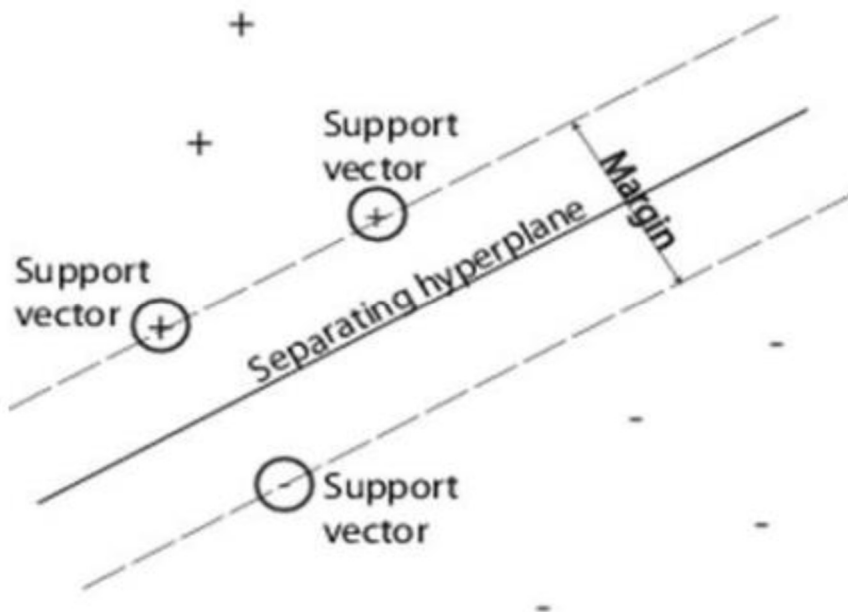
Jonkerin ja muiden [2017] mukaan käänteisellä myrkytyksellä voi olla tuhoiset seuraukset, jos sitä sovelletaan vaalituloksien manipulointiin: järjestelmää voi huijata käänteismyrkyttämällä suodattimen niin, että tärkeät

viestit ajautuvat käyttäjien spam-kansioon. Kokeiden perusteella pidettiin mahdollisena, että jopa Gmailin roskapostisuodattimet voisi tällä tavoin käänteismyrkyttää luokittelemaan vaaleihin liittyviä sähköposteja roskapostiksi.

Koska monissa palveluissa käyttäjät jakavat sähköpostipalveluntarjoajan kanssa yhteisen suodattimen, myrkyttäjän riittäisi lähettää pelkästään itselleen viestejä estääkseen tärkeiden viestien kulkeminen oikeaan kansioon koko palvelussa.

## 5.2 Tukivektorikone

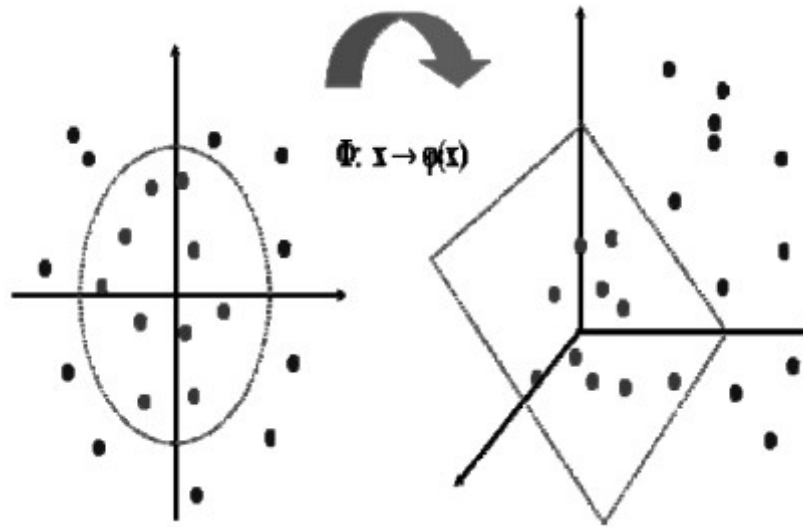
Tukivektorikone (SVM) on yksi käytetyimmistä tekniikoista. SVM pyrkii piirtämään optimaalisen hypertason luokkien välille ja erottamaan luokat toisistaan niin, että hypertaso on mahdollisimman kaukana kummankin luokan datasta, eli luokkien välinen marginaali maksimoidaan. Roskapostisuodatuksessa SVM etsii siis piirreavaruudesta spam- ja ham-luokkien väliin piirrettävää optimaalista hypertasoa, jonka toiselle puolelle jäävät roskapostiviestit ja toiselle puolelle hyödylliset viestit. Marginaali määritellään käyttämällä luokkien rajalla sijaitsevia reunimmaisista pisteistä tukivektoreina (Kuva 6). Piirreavaruudessa sijaitsevat pisteet ovat tässä tapauksessa datan sähköpostiviestejä.



Kuva 6. Hypertaso (hyperplane) erottaa luokat toisistaan tukivektorien (support vector) välissä. [Shradhanjali and Verma 2017].

Mikäli hypertasoa ei voida piirtää lineaarisesti, voidaan lisätä syöteavaruuteen ulottuvuuksia ottamalla mukaan lisää piirteitä ja näin muuntaen sen korkeadimensionaalisemmaksi piirreavaruudeksi. Muunnos korkeamman dimensionaalisuuden avaruuteen on tarpeellinen, sillä lineaarinen operaatio

piirreavaruudessa on yhtä kuin epälineaarinen operaatio syöteavaruudessa (Kuva 7). Tätä kutsutaan myös kernel-tempuksi.



Kuva 7. Epälineaarisen ongelman ratkaisemiseksi voidaan muuntaa syöteavaruus korkeamman ulottuvuuden piirreavaruudeksi [Chhabra *et al.* 2010].

Tukivektorikonetta määrittää suurelta osin siihen valittu kerneli eli ydinfunktio. Ydinfunktioiden päämuodot ovat lineaarinen ydinfunktio, polynomiaalinen ydinfunktio, RBF (radial basis function) ja sigmoidifunktio (sigmoid function). [Chhabra *et al.* 2010].

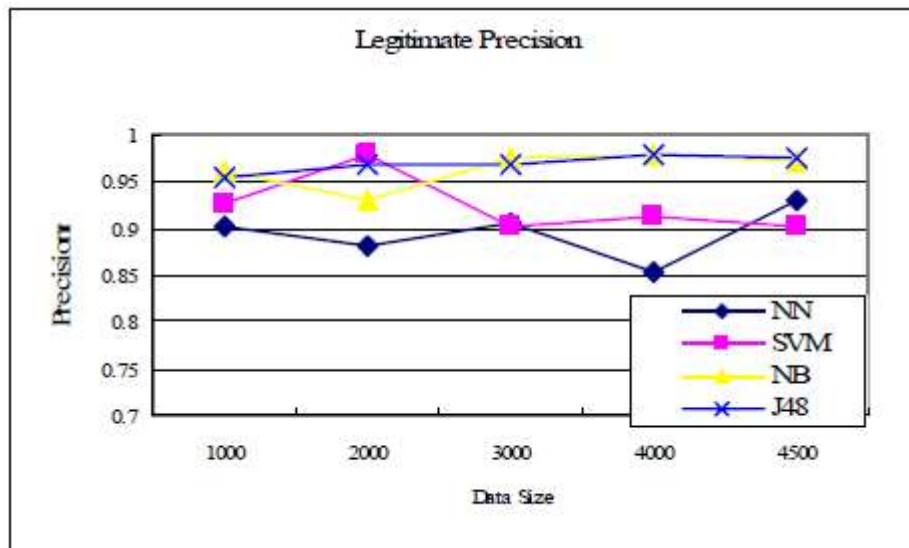
Vahvuuksiensa ansiosta SVM on hyvin suosittu. SVM on tehokas, kun käsitellään suurta määrää piirteitä [Sah and Parmar 2017]. Joachimsin [1998] mukaan tekstiluokittelussa joudutaan käsittelemään jopa yli kymmenentä tuhatta piirrettä. Koska SVM käyttää ylisovitusta (overfitting) vastaan suojausta, joka ei välttämättä riipu piirteiden määrästä, on sillä potentiaalia käsitellä suuria piirreavaruuksia. Yksi tapa välttää liiallista dimensionaalisuutta on olettaa, että suurin osa piirteistä on epärelevantteja. Lisäksi suurin osa tekstiluokittelusta on lineaarisesti erotettavissa luokkiin.

Hyödyntämällä kernel-funktiota SVM pystyy tehokkaasti ratkaisemaan dimensionaalisuuden ongelman, ja on osoittanut erinomaista yleistyskykyä korkeadimensionaalisessa avaruudessa [Chhabra *et al.* 2010]. Tarkkuus on hyvä, jopa 97,99% [Daeeff *et al.* 2016].

SVM:n heikkouksia on suurien datamäärien luokittelu, jossa SVM ei ole kovin tehokas, ja kuluttaa paljon aikaa ja muistia [Chhabra *et al.* 2010]. SVM vaatii paljon laskentatehoja opetteluvaiheessa, ja menetelmä on myös taipuvainen ylisovittamiseen silloin, kun datassa ilmenee paljon kohinaa [Abu-Nimeh *et al.* 2007].

## 6. Vertailu

Edellä esitettyjen vahvuuksien ja heikkouksien perusteella naiivi Bayes-luokittelija ei ole aina yhtä tarkka kuin tukivektorikone, mutta sietää paremmin suurta dataa ja poikkeavuuksia datassa. Myös Younin ja McLeodin [2006] mukaan tukivektorikoneen täsmällisyys kärsii suurissa määrissä dataa eikä se suoriutunut yhtä hyvin kuin naiivi Bayes-luokittelija, joskin pienemmissä datajoukoissa tukivektorikone taas pärjäsikin testissä paremmin (Kuva 8.)



Kuva 8. NB (keltainen kolmio) = naiivi Bayes, SVM (vaaleanpunainen neliö) = tukivektorikone. Muut viivat ovat tutkimuksessa mukana olleet neuroverkko (tumman sinisellä) ja J48 (vaaleansinisellä). [Youn and McLeod 2006]

Naiivi Bayes-luokittelija on yksinkertaisempi ja vaatimuksiltaan kevyempi kuin tukivektorikone, mutta altis korkean dimensionaalisuuden kiroukselle ja bayesilaiselle myrkytykselle.

Tukivektorikone kykenee käsittelemään pienempää datajoukkoa hyvin tarkastikin ja toimii hyvin korkeadimensionaalisessakin avaruudessa, kun taas naiivi Bayes-luokittelijalle piirreavaruuden suuri dimensionaalisuus on ongelma.

Yleisesti tehokkaimpana pidetyn, suosituksen ja tarkan tukivektorikoneen tarkkuus riippuu paljon oikean kernelin valinnasta, kun taas naiivi Bayes-luokittelijan tarkkuus terminvalikoimismenetelmästä.

Kummassakin suodatinmenetelmässä oikeiden piirteiden oikea määrä ja valikointi ovat kriittisiä toiminnan kannalta. Olennaista kummankin menetelmän kannalta siis on, sopiiko suodatinmenetelmä sopiva käsiteltävään ongelmaan, datajoukon kokoon laatuun.

## 7. Johtopäätökset

Tässä tutkielmassa pyrittiin tarkastelemaan roskapostisuodattimia koskevaa tutkimusta, löytämään tutkimustrendien perusteella kaksi alan suosittua ja tehokasta suodatinmenetelmää joita voi pitää merkittävinä, vertailla niitä ja erotella niiden ominaisuuksia.

Mukaan otettujen tutkimusten perusteella tukivektorikone ja naiivi Bayes-luokittelija sopivat erilaisiin tilanteisiin: tukivektorikone tarjoaa tarkkuutta, mutta vaati laskentatehoa ja aikaa, ja naiivi Bayes-luokittelija vaati vähemmän tehoja mutta ei ole aina yhtä tarkka. Tukivektorikone on suosionsa perusteella paras, ja binäärisessä roskapostisuodatuksessa data on usein sopivasti lineaarisesti erotettavissa.

Lisäksi ilmeni potentiaalisia vaaroja tietoturvan ja jopa ihmisoikeuksien kannalta. Käyttäjä luottaa tärkeän elämän osa-alueen, eli sähköpostiviestien suodattamisen, tekoälyn käsiin.

Koska on mahdollisuus, että kokonaisen palvelun suodattimen väärä toiminta voisi pahimmillaan estää tärkeiden viestien pääsemisen perille, kuten edellä mainittu käänteisen bayesilaisen myrkytyksen mahdollinen uhka osoittaa, ansaitsee se näin ollen huomiota tulevaisuuden tutkimuksessa.

Koska häiritsevien mainosten ja muun roskapostin lisäksi käyttäjän toinen uhka ovat mahdolliset tahot, jotka saattavat haluta manipuloida suodattimen toimintaa ja hyväksikäyttää sitä, asettaa se roskapostisuodattimien toiminnalle tiettyjä eettisiä vaatimuksia, ja osoittaa, että suodattimen vastuu tärkeiden asiakirjojen käsittelijänä on suuri.

Käyttäjän kannalta oikean suodatintekniikan valitseminen tilanteeseen sopivasti oikeine vahvuuksineen ja heikkouksineen sekä tarkkuuden ja virheiden laatu ja määrä ovat siis kriittisiä ymmärtää.

Eräs tulevaisuuden kehitettävistä piirteistä roskapostisuodattimissa onkin väärrien positiivisten spam-luokitusten määrän vähentäminen, sillä jos suodatin hävittää käyttäjän tärkeinä pitämiä viestejä, voi käyttäjälle tulla väärästä positiivisesta luokittelusta silloin enemmän vahinkoa kuin suodattamatta jääneistä spam-viesteistä, jos kyseessä on esimerkiksi suuren yrityksen asiakirjoista [Mujtaba *et al.* 2017].

Tämän tutkielman rajoitteina olivat sen pieni otos tutkimuksia ja niiden painottuminen tutkimuksiin, jotka ovat helposti saatavilla yleisistä tietokannoista ja erityisesti Google Scholarista.

Tässä tutkielmassa ei käsitelty juurikaan moniluokitusta tai eri luokittelumenetelmien yhdistelyä, vaikka ne ovat tulevaisuuden tutkimusten ja menetelmien kannalta kiinnostavia. Esimerkiksi tukivektorikoneen yksi heikkous ilmenee Liun ja muiden [2005] mukaan juuri moniluokittelutapauksissa,



joissa on monia äärimmäisen harvinaisia kategorioita ja poikkeuksia, ja tukivektori-koneen tarkkuus oli tällöin huono.

Luokittelumenetelmiä yhdistelemällä niiden heikkouksia voidaan kompensoida ja vahvuuksia hyödyntää yhdessä: esimerkiksi Daeefin ja muiden [2016] mukaan MSPEDS-järjestelmällä (multi stage phishing email detection system), jossa naiivi Bayes-luokittelija on yhdistettynä monivaiheisessa luokituksessa päätöspuihin ja satunnaismetsä-tekniikkaan, saadaan väärin positiivisten ja väärin negatiivisten luokittelujen todennäköisyys putoamaan jopa 0.4 prosenttiin. Järjestelmän suoritusajan vähentämisessä on tosin vielä kehitettävää.

Menetelmät, jotka nykypäivänä kärsivät suurista tehovaatimuksista, saattavat tulevaisuudessa olla merkittäviä, sillä vuosien mittaan menetelmiä optimoidaan tehokkaammiksi ja tietotekniikan kehittyessä jotkin muistivaatimukset ja muut hidastavat tekijät saattavat uuden teknologian myötä lakata olemasta ongelma.

Muita roskapostisuodatusmenetelmiä kuten päätöspuita tai neuroverkkoa ei käsitelty, vaikka niistä on olemassa kiinnostavaa tutkimusta. Suorituskyvyn vertailussa keskityttiin vain pääpiirteisiin, joita tutkimuksissa mainittiin, ja kuvailtiin suorituskykyä muutamilla yleisillä kriteereillä, menemättä juuri tarkempiin teknisiin selityksiin esimerkiksi suodattimien suorituskyvyn eroista.

Laajemmassa tutkimuksessa olisi voitu ottaa mukaan kyberturvallisuuden ja massavalvonnan ja jopa kyberterrorismin näkökulmaa, ja pohtia eri suodattimien luotettavuutta ja sopivuutta eri uhkiin tarkemmin.

## Viiteluettelo

- Saeed Abu-Nimeh, Dario Nappa, Xinlei Wang and Suku Nair. 2007. A comparison of machine learning techniques for phishing detection. In: *Proc. of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit*, 60-69.
- Tiago A. Almeida, Jurando Almeida and Akebo Yamakami. 2011. Spam filtering: How the dimensionality reduction affects the accuracy of naive-Bayes classifiers. *Journal of Internet Services and Applications* 1, 3, 1-2.
- Ion Androutsopoulos, Georgios Paliouras and Eirinaios Michelakis. 2004. Learning to filter unsolicited commercial e-mail. *Technical Report 2004/2*. National Centre for Scientific Research "Demokritos".
- Priyanka Chhabra, Rajesh Wadhvani and Sanyam Shukla. 2010. Spam filtering using support vector machine. *IJCCT* 1(2, 3, 4), 166-171.
- Ammar Yahya Daeef, R. Badlishah Ahmad, Yasmin Yacob, Naimah Yaakob and Ku Nurul Fazira Ku Azir. 2016. Multi stage phishing email classification, *J. Theor. Appl. Inf. Technol.* 83, 206-214.

- Masooome Esmaili, Arezoo Arjomandzadeh, Reza Shams and Morteza Zahedi. 2017. An anti-spam system using naive Bayes method and feature selection methods. *International Journal of Computer Applications* 165, 4, 4.
- Sakshi Hooda, Aakansha, Varsha Kansal, and Swati Kadian. 2015. Comparison and analysis of spam detection algorithms. *International Journal for Research in Applied Science & Engineering Technology* 3, 1, 21-22.
- Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In: *Proc. of the European Conference on Machine Learning*, 137-142.
- Hugo Jonker, Sjouke Mauw and Tom Schmitz. 2017. Reverse Bayesian poisoning: How to use spam filters to manipulate online elections. In: *Proc. of the 2nd International Joint Conference on Electronic Voting (E-Vote-ID'17), Lecture Notes in Computer Science*, 10615, 183-197.
- Tie-Yan Liu, Yiming Yang, Hao Wan, Hua-Jun Zeng, Zheng Chen, and Wei-Ying Ma. 2005. Support Vector Machines classification with a very large-scale taxonomy. *SIGKDD Explorations* 7, 1, 36-43.
- Mehryar Mohri, Afshin Rostamizadeh and Ameet Talwalkar. 2012. *Foundations of Machine Learning*. The MIT Press.
- Ghulam Mujtaba, Liyana Shuib, Ram Gopal Raj, Nahdia Majeed and Mohammed Ali Al-Garadi. 2017. Email classification research trends: Review and open issues, *IEEE Access* 5, 9059.
- Umesh Kumar Sah and Narendra Parmar. 2017. An approach for malicious spam detection in email with comparison of different classifiers. *International Research Journal of Engineering and Technology* 04, 08, 2238-2242.
- Mehran Sahami, Susan Dumais, David Heckerman and Eric Horvitz. 1998. A Bayesian approach to filtering junk e-mail. *AAAI Tech. Rep. WS-98-05*, 1-2.
- Shradhanjali and Toran Verma. 2017. E-mail spam detection and classification using svm and feature extraction. *International Journal of Advance Research, Ideas and Innovations in Technology* 3, 3, 1491-1495.
- Fagbola Temitayo, Olabiyisi Stephen and Adigun Abimbola. 2012. Hybrid GASVM for efficient feature selection in e-mail classification. *Computer Engineering and Intelligent Systems* 3, 3, 17- 28.
- Seongwook Youn and Dennis McLeod. 2007. A comparative study for email classification. In: Khaled Elleithy (ed.), *Advances and Innovations in Systems, Computer Sciences and Software Engineering*. Springer, 387-391.

# Informaation visualisoinnista big dataan: katsaus kirjallisuuteen

Juuso Kuisma

## Tiivistelmä.

Jotta visualisointi tarjoaisi oivalluksia ja tukea ihmisen kognitiolle, sen tulisi olla sekä tehokas että hyödyllinen käyttökontekstissaan. Tämä vaatii yhdistelmän suunnittelua ja tiedettä, jotta muutoin vaikeaselkoinen tieto paljastuu tutkijalleen. Tutkielmassa käsitellään kirjallisuuden kautta informaation visualisointia yleisellä tasolla, jonka jälkeen perehdytään tarkemmin visualisoinnin haasteisiin, strategiaan ja vakiintuneisiin visualisointimethodeihin big datan käyttökontekstissa.

**Avainsanat ja -sanonnat:** informaation visualisointi, big data, visualisointimenetelmät.

## 1. Johdanto

Tulevaisuuden harppaukset tieteessä, tekniikan alalla ja lääketieteessä ovat riippuvaisia taidosta ymmärtää laajoja informaatiomääriä. Informaation visualisointi on tärkeä avain, joka tekee pyrkimyksen mahdolliseksi: se auttaa ihmisiä tutkimaan ja selittämään tietoa ohjelmistojen avulla, jotka tuottavat staattisia tai vuorovaikutteisia visuaalisia esityksiä tiedosta. Vaikka informaation visualisointi vaikuttaa lupaavalta edesauttajalta eri tieteenaloilla, teknologian soveltaminen käytännössä ei ole yksinkertaista. Tehokkaiden visualisointimethodien kehittäminen on monimutkainen prosessi, joka vaatii ymmärrystä sekä olemassa olevista metodeista että siitä, kuinka metodit vastaavat ihmisen kognitioon.

Tutkielma pyrkii tarkastelemaan kirjallisuuden kautta, mitä informaation visualisointi yleisesti tarkoittaa, jonka jälkeen tutkielmassa paneudutaan syvällisemmin informaation visualisointiin big datan käyttökontekstissa. Tutkielma pyrkii esittelemään ilmiötä yleisesti, ja kuinka suuret aineistomäärät muodostavat omia haasteitaan informaation visualisoinnille. Myöhemmin tutkielmassa perehdytään strategiaan big datan visualisoinnissa ja arvioidaan, kuinka vakiintuneet visualisointimethodit soveltuvat big datan ominaispiirteisiin.

## 2. Informaation visualisointi

Informaation visualisoinnilla tarkoitetaan pitkälti tietokoneavustettua, interaktiivista tapaa tehdä visuaalisia esityksiä tiedosta avustamaan kognitiota. Kognitiolla tarkoitetaan ihmisen sisäisiä mentaalisia prosesseja, jotka liittyvät sekä

tiedonhankintaan että vastaanotetun tiedon prosessointiin. Tiedon visualisoinnin tarkoitus on auttaa tiedon syvemmässä ymmärryksessä ja oivaltamisessa. Tiedon syvällisempi ymmärrys taas tuottaa uusia löydöksiä, avustaa päätöksenteossa ja tarjoaa selityksiä erilaisille ilmiöille. [Card *et al.* 1999.]

Informaation visualisointia voi Chenin [2006] mukaan tarkastella kahden eri näkökannan kautta, joista ensimmäinen käsittää informaation (1) rakenteellisen mallintamisen ja toinen käsittää informaation (2) graafisen esityksen. Seuraavat kappaleet pyrkivät avaamaan lyhyesti esimerkkien avulla, mitä kahdella eri näkökulmalla tarkoitetaan käytännössä.

Rakenteellisen mallintamisen tarkoitus on pääsääntöisesti havaita, määrittää ja yksinkertaistaa tiedon alla olevia suhteita. Tietojen suhteet toisiinsa muodostavat rakenteen, joka kuvaa dataa kokonaisuutena. Rakenteellinen mallintaminen pyrkii vastaamaan kysymyksiin tiedon luonteesta, kuten esimerkiksi mikä on monimutkaisen tietoverkoston tai dokumenttikokoelman rakenne. Käytännössä rakenteellista mallintamista on myös selvitys siitä, minkälaisia mentaalisia malleja ihmisillä on esimerkiksi tilastosta, rakennuksesta tai vaikkapa kaupungista.

Informaation graafisella esityksellä on toisaalta tarkoitus muodostaa rakenteellisesta mallista visuaalinen esitys, jota ihminen voi tutkia, ja jonka kanssa ihminen voi olla vuorovaikutuksessa. Esimerkiksi hierarkkinen suhde tietojen välillä voitaisiin esittää visuaalisesti puumaisella kaaviolla, jossa lukujärjestys ylhäältä alas alleviivaa tiedon hierarkkisuutta. On hyvä huomata, että luokittelun rajat ovat arkikielessä usein häilyvät. Usein puhuttaessa informaation visualisoinnista viitataan vain toiseen näkökulmaan aiheesta, vaikka aiheella on kaksi toisiinsa kiinteästi liittyvää näkökulmaa.

Informaation visualisoinnista on selkeitä hyötyjä, kun halutaan tukea ihmisen kognitiivisia prosesseja, kuten esimerkiksi tiedonhakua. Tiedon visualisointi on erityisen arvokasta siksi, että ihminen on harjaantunut huomaamaan erilaisia malleja ja kaavoja käsitellessään tietoa [Card *et al.* 1999], joten etenkin visualisoinnin avulla mallinnetut muutokset tiedoissa ovat tehokkaita välittämään tietoa.

Seuraavaksi esitellään tiedon visualisoinnin tarjoamia merkittäviä hyötyjä. Tutkielmassa esitettiin aikaisemmin, kuinka tiedon visualisointi avustaa ajattelua ja kuinka visualisoinnilla on tärkeä kytkös ihmisen kognitiivisiin toimintoihin. Card ja muut [1999] esittävät, että tietokoneavustetulla visualisoinnilla, esimerkiksi tiedonhaussa ja hakutulosten visuaalisessa esittämisessä eri tietokannoista tai -varastoista, on erityistä hyötyä viidestä eri pääsyystä:

1. Visualisointi antaa käyttäjälle lisää resursseja tiedon hahmottamiseen ja laajentaa käyttäjän käytössä olevaa rajoitettua työmuistia.
2. Visualisoinnilla voidaan vähentää tiedon etsintään käytettyä aikaa.
3. Käyttäjän on helpompi tunnistaa tiedossa esiintyviä malleja ja kaavoja.
4. Käyttäjä voi päätellä tietoon liittyviä asioita aistinvaraisesti, etenkin jos tieto päivittyy tai muuttuu reaaliaikaisesti.
5. Visualisointia itsessään voi manipuloida ja muuttaa vastaamaan paremmin tarpeita.

Esteettisellä, toimivalla visualisoinnilla on myös aikaisemman viiden hyödyn lisäksi ylimääräinen merkittävä hyöty: estetiikan on osoitettu kohentavan tehokkuutta suoritettavassa tehtävässä vähentäen samanaikaisesti virheiden määrää. [Cawthon and Moere 2007.] Chenin [2006] mukaan esteettinen visualisoinnin suunnittelu on yksi merkittävimmistä ongelmista informaation visualisoinnin saralla. Visuaalisesti miellyttävä näkymä vetää käyttäjää puoleensa ja voi toimia kannustimena järjestelmän käytön aloittamisessa, mutta tutkijoille ei ole vielä täysin varmaa, miksi tietyt visualisoinnit ovat viehättävämpiä kuin toiset.

### **3. Big data**

Asiakkaiden tarve toissijaisen datan käsittelylle, joka ei siis ole suoraan kytköksissä asiakkaan yritystoimintaan, on luonut modernin ilmiön, josta käytetään kirjoittamisen aikaan tietojenkäsittelyn alalla vakiintuneesti nimitystä big data. Termi viittaa ilmiöön, jolla ei ole tarkkaan asetettuja rajoja. Big dataa voidaan luonnehtia loputtomalla määrällä kerääntyvää tietoa. Tiedon kertymää voidaan myös esittää lukuisissa eri tiedon tallennusformaateissa, joista monet eivät ole rakenteellisia tietovirtoja. [Blue 2011.]

Yleensä big data ymmärretään suurina aineistoina, joiden koko kasvaa eksponentiaalisesti. Tämä aineisto saattaa olla liian suurta, käsittelemätöntä tai liian jäsentelemätöntä perinteisille tietojenkäsittelyn metodeille, joita käytetään relaatiotietokantojen yhteydessä. Suurin huolenaihe ei kuitenkaan ole aineiston suuruus ja datan määrä, vaan kuinka aineistoa voidaan soveltaa ja käyttää hyödyksi. [Blue 2011.]

Big datalla on seuraavat ominaisuudet tai tunnuspiirteet: aineistoa on suuri määrä (Volume), dataa voidaan esittää monissa eri formaateissa (Variety) ja datan käsittelyn nopeus ja dynaamisuuden taso ovat korkeita (Velocity). Mikäli arvioitavana oleva aineisto täyttää kaksi kolmesta mainitusta tunnuspiirteestä, voidaan aineistoa luonnehtia big dataksi. Kolmen tunnuspiirteen perusteella big datalle on ehdotettu luokitteluja, joihin kuuluvat "Volume-Velocity" -luokka, "Volume-Variety" -luokka, "Velocity-Variety" -luokka, sekä kaikki kolme

ominaisuutta käsittävä "Volume-Velocity-Variety" -luokka. [Dijks 2013; Zikopoulos 2012.]

Big datan käsittely ei ole yksinkertainen tehtävä, vaan aineiston käsittely vaatii erityisiä metodeja ja lähestymistapoja. Visualisointiin perustuva tiedon prosessointi on ihmisille luontainen tapa käsitellä informaatiota, joten voidaan sanoa, että graafinen esitys aineistosta on tehokas metodi, joka auttaa ihmistä sekä ymmärtämään aineistoa paremmin että tarjoaa tukea ihmisen kognitiivisille prosesseille, jotta päätöksien tekeminen helpottuu. Kuitenkin big datan ollessa kyseessä, monet perinteiset datan esitysmenetit eivät ole yhtä tehokkaita tai käytännöllisiä konkreettisille tehtäville [Zikopoulos 2012].

Vaikka informaation visualisoinnista on yleisesti merkittäviä hyötyjä, big datan kontekstissa visualisoinnissa on havaittavissa ongelmia. Seuraavat kappaleet esittelevät lyhyesti big datan visualisointiin liittyviä ongelmia. Ongelmat visualisoinnissa juontuvat big datan ominaisuuksista, jotka esiteltiin aikaisemmin tutkielmassa.

Graafinen analyysi ei rajoitu pelkästään staattisiin kuvallisiin visualisointeihin. Dynaamiset muuttuvat visualisoinnit asettavat omia haasteitaan ja vaativat vaihtelevan määrän suorituskykyä tietokoneelta. Tutkielman seuraavissa alakohdissa esitettävät ongelmat korostuvat entisestään luonnoltaan dynaamisissa käyttöympäristöissä, joissa nopea suorituskyky on tärkeää.

Eräs merkittävä ongelma big datan visualisoinnissa on niin kutsuttu visuaalinen kohina. Visuaalisella kohinalla tarkoitetaan käytännössä sitä, että esimerkiksi näyttöpäätteellä esitetty dataa sisältävä taulukko on vaikeaselkoinen tiedon määrästä johtuen. Taulukon tietueet ovat joko liian lähekkäin, tietueita on liikaa tai tietueita on vaikea erottaa toisistaan. Taulukon tarkastelijan on vaikeaa saada selkoa visualisoidusta tiedosta ilman esikäännöksen yhteydessä suoritettua aineiston prosessointia. Liiallinen visuaalinen kohina aiheuttaa ongelmia tiedon näkyvyyden kanssa. [Agrawal *et al.* 2015; Kirk 2012.]

Yksinkertainen ratkaisu visuaaliselle kohinalle on suurempien näyttöpäätteiden käyttäminen datan visualisoinnin yhteydessä. Tilan lisääminen voi kuitenkin aiheuttaa uuden ongelman, joka liittyy ihmisen luontaiseen havaintokykyyn. Vaikka suurien kuvien visualisoinnin taso on paljon korkeampi verrattuna esimerkiksi taulukoituun visualisointitapaan, suuret kuvat voivat ylikuormittaa ihmisen luontaista havaintokykyä, jolloin kuvasta on vaikeaa saada irti tarvittavaa tietoa. [Agrawal *et al.* 2015; Kirk 2012.] Big datan kontekstissa jatkuvasti kasvavat samanaikaisesti esitettävät tietomäärät hankaloittavat visualisointia entisestään [Zikopoulos 2012]. Näin ollen voidaan sanoa, että datan visualisointiin käytettyjä metodeja rajoittavat sekä näyttöpäätteen kuvasuhde ja resoluutio sekä ihmisen fyysisen havaintokyvyn asettamat rajoitteet.

Aikaisemmin esitettyihin ongelmiin ratkaisuksi voidaan käyttää lähestymistapaa visualisointiin, jossa tietoa kootaan yhteen ja ryhmitellään erilaisilla suodatusmenetelmillä, jolloin tietueita tai olioita aineistoissa yhdistetään eri kriteerien perusteella. Ongelma on kuitenkin siinä, että yhdistely voi johtaa informaation häviöön. [Agrawal *et al.* 2015; Kirk 2012.] Visualisoitavien aineistojen suodattaminen voi olla harhaanjohtavaa, sillä merkittävä informaatio suurissa tietomäärissä voi hävitä näkyvistä suodattamisen yhteydessä. Monimutkainen yhdistäminen voi lisäksi vaatia paljon aikaa ja resursseja suorituskyvyltä, jotta aineistosta saadaan riittävän tarkkaa tutkijan vaatimaa tietoa [Tedesco *et al.* 2013].

Viimeinen esitettävä ongelma on erityisen merkittävä tilanteissa, joissa aineisto vaatii jatkuvaa monitorointia, jolloin tietovirtaa tarkkaileva henkilö ei välttämättä kykene reagoimaan tiedon muutoksien määrään tai siihen, kuinka paljon tietoa esitetään näyttöpäätteellä. Muuttuvuusasteen hidastaminen ei välttämättä korjaa esiintyvää ongelmaa, sillä muuttuvuusaste on riippuvainen tarkkailijan reaktionopeudesta ja harjaantuneisuudesta tehtävässä. [Tedesco *et al.* 2013.] Kaiken kaikkiaan big datan visualisointiin liittyvät ongelmat johtavat lopputulokseen, jossa tutkittavan aineiston analyysin laatu laskee [Agrawal *et al.* 2015].

### **3 Lähestymistapoja big datan visualisointiin**

Useimmat tiedon visualisointimetodit eivät kehity tyhjiössä, vaan rakentuvat jo kehitettyjen metodien varaan [Card *et al.* 1999; Chen 2006]. Tämä luku keskittyy big datan visualisointiin strategisesta näkökulmasta, jonka jälkeen esitellään konkreettisia visualisointimetoja omassa luvussaan.

Graafista visualisointia käytetään monimuotoisesti eri sovellusaloilla, mutta tietomäärien kasvaessa visualisoinnin tehokkuus ja käyttökelpoisuus muodostuvat ongelmaksi. Haaste on kaksijakoinen: (1) visualisoitava data pitäisi pystyä paloittelemaan osiksi, jotka esitetään omina osakokonaisuuksinaan ja (2) visualisoinnin pitäisi pystyä vastaamaan eri tekijöiden asettamiin rajoitteisiin. [Card *et al.* 1999.]

Kirkin [2012] mukaan vastaamalla näihin haasteisiin voidaan luoda lähestymistapa, joka tuottaa havainnollisempia ja informatiivisempia esityksiä, jotka lopulta auttavat löytämään piilotettuja yhteyksiä tutkittavista aineistoista.

Lähestyttäessä big datan visualisointia analyysiin käytettävien työkalujen tulisi tukea seuraavia toimintoja, joita käsitellään tarkemmin seuraavissa kohdissa:

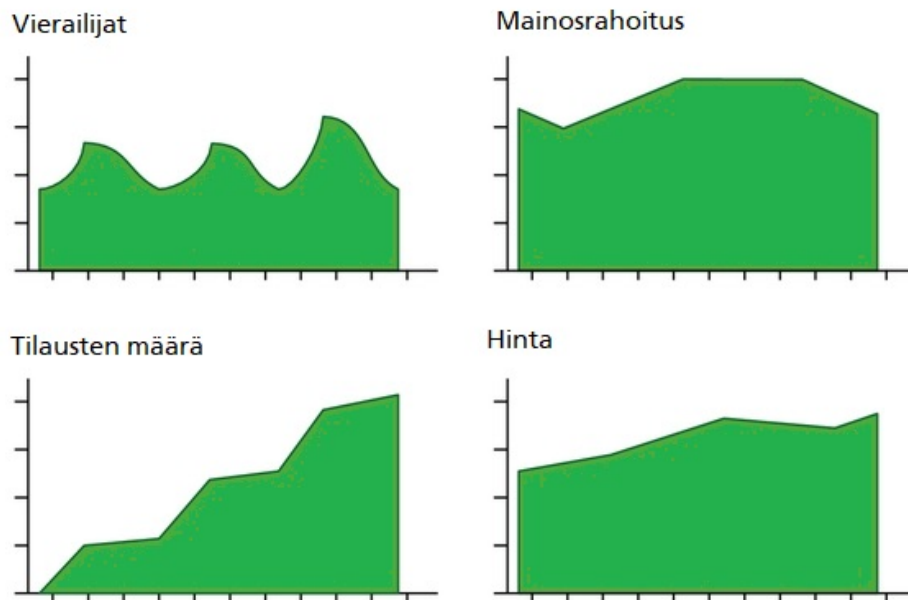
1. Käyttäjän tulisi pystyä tarkastelemaan aineistoa useamman kuin yhden näkymän kautta.

2. Käyttäjän pitäisi pystyä olemaan aktiivisesti vuorovaikutuksessa analysoitavan näkymän kanssa.
3. Käyttäjän pitäisi pystyä muuttamaan eri tekijöitä dynaamisesti työskentelyprosessin aikana näkymän parissa.

### 3.1 Usean näkymän esitys näyttöpäätteellä

Ymmärtääkseen täysin tutkittavaa aineistoa tutkijat käyttävät tavallisesti yksinkertaista lähestymistapaa, jossa käytetään perinteisiä näkymiä tiedoista, jotka sisältävät rajoitetun määrän osatekijöitä, jotta aineistosta on helppo tunnistaa yhteyksiä joko näkymien välillä tai yhdessä konkreettisesti näkymässä. [Robertson *et al.* 2008; Heer and Shneiderman 2012.]

Vaikka käytettävissä on kokonainen joukko visualisointimetodeja, usein nähdään käytettävän vain samankaltaisia tai lähes samanlaisia objekteja. Esi-merkki tästä on viiva- tai pistediagrammi (Kuva 1). Aineiston tarkastelija saattaa olla kiinnostunut monista eri tavoista visualisoida samaa dataa, mutta samalla visuaalisen analyysin prosessi kokonaisuudessaan vaikeutuu. Tässä tilanteessa tarkastelijan pitäisi pystyä vertailemaan sekä samankaltaisia graafisia objekteja että kyetä erottamaan eri tiedot toisistaan eri tekijöiden perusteella. [SAS Institute 2017.]



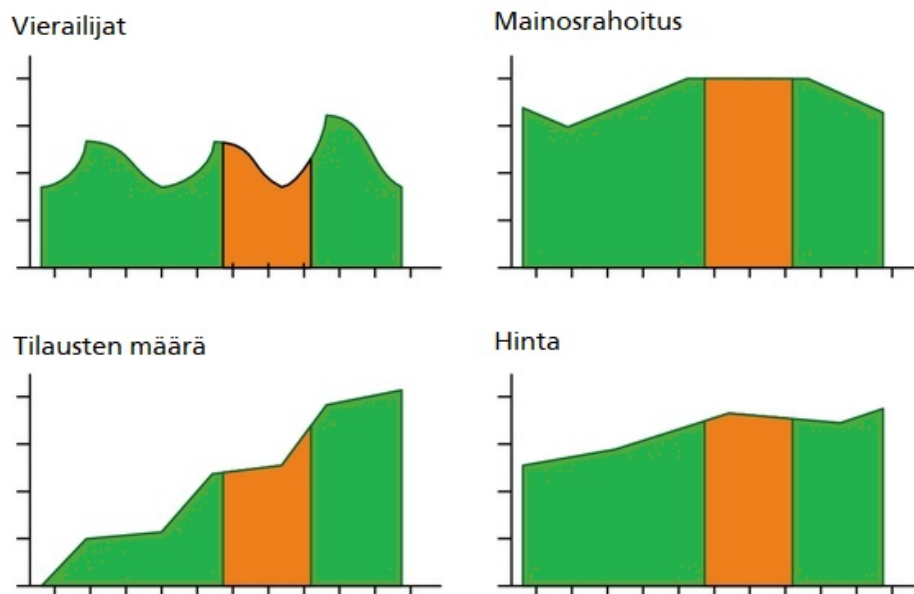
Kuva 1. Neljän visualisoinnin esitys samassa näkymässä

Seurauksena lähestymistavasta voidaan sanoa, että aineiston tarkastelija saattaa ohjautua toivottavaan lopputulokseen jo tutkimusprosessin alkuvaiheessa, sillä perinteiset näkymät, kuten diagrammit, tukevat johtopäätösten tekoa. Joissakin tapauksissa yksinkertainen lähestymistapa usean näkymän esi-



tyksellä voi tarjota tutkimukselle lopputuloksen, koska aineiston tarkastelija välttyy ajautumasta tutkimustyön kannalta vääriin lopputuloksiin.

Lisäksi tärkeä avain lähestymistavassa on mahdollisuus valita halutut alueet datasta kaikkiin tietoihin liittyviin esityksiin, kuten kuva 2 osoittaa.



Kuva 2. Valintojen teko visualisoidusta datasta

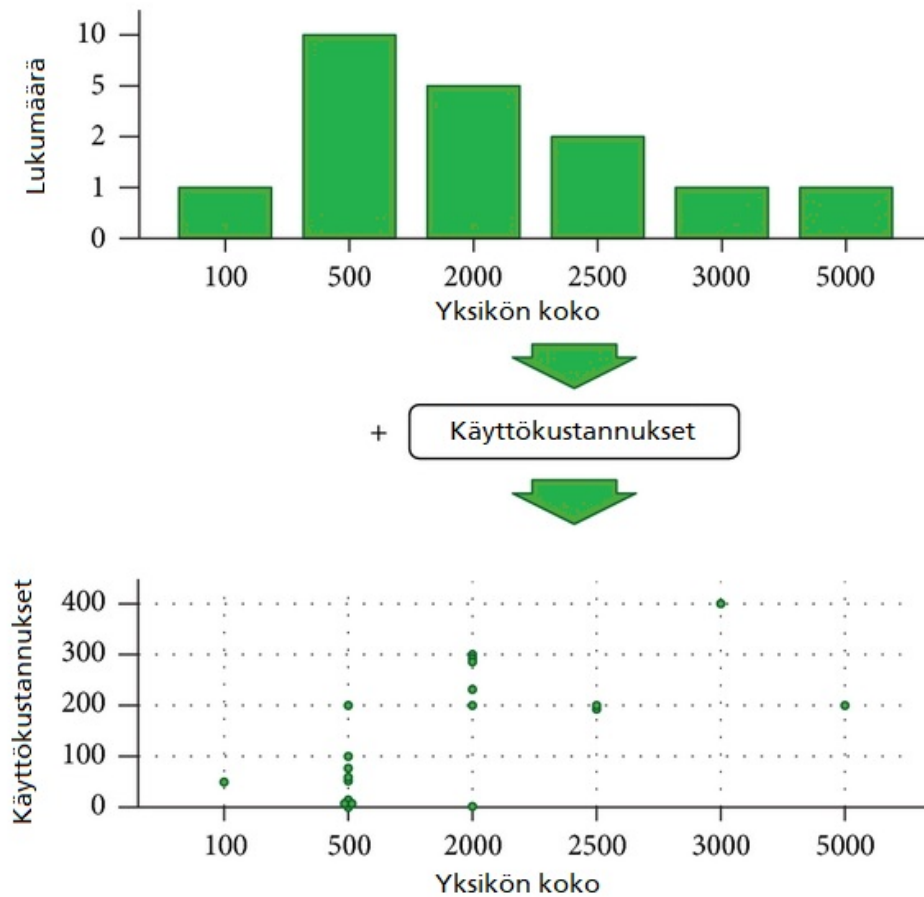
Aineiston tarkastelijat saattavat myös haluta koordinoida näkymiä monin eri tavoin, kuten tilanteissa, joissa yhden näkymän sisällä tehdyt valinnat korostavat toisessa näkymässä havaittavia yhteen liittyviä tietueita. Vastaavasti yhdessä näkymässä tietojen suodattaminen voi häivyttää toissijaista tietoa näkyvistä toisessa näkymässä. Yhteen linkitetty navigointi näkymien välillä voi myös tarjota tukea koordinoinnille, kuten tilanteissa, joissa toisessa näkymässä vierittäminen tai zoomaaminen muuttaa samanaikaisesti muita näkymiä. [Heer and Shneiderman 2012; Cleveland and McGill 1984.]

### 3.2 Vuorovaikutus näkymän kanssa

Yksi tärkeimmistä perustavanlaatuisista operaatioista visuaalisessa analysoinnissa on vuorovaikutus tutkittavan näkymän kanssa. Tarkastelijan pitää pystyä määrittelemään, mikä tieto esitetään ja kuinka tieto tulisi esittää, jotta visualisointi avustaa havaintojen tekoa.

Kutakin graafista visualisointitapaa voidaan käyttää hyväksi mihinkä tahansa tietoon, mutta onko valittu metodi välttämättä hyödyllinen käytettävän informaation löytämisessä? Tyypillisesti big datan ollessa kyseessä tarkastelija ei pysty havainnoimaan koko aineistoa kerralla tai löytämään aineistoista poikkeamia tai suhteita ensisilmäyksellä [SAS Institute 2017]. Ajankohtainen lähes-

tymistapa on osatekijöiden lukumäärien muuttaminen dynaamisesti [SAS Institute 2017; Panopticon Software 2017]. Esimerkiksi tilanteessa, jossa aineiston tarkastelija on valinnut osatekijän, tarkastelija näkee perinteisen pylväsdiagrammin, joka näyttää tietueiden jakautumisen tietuetyypin mukaan. Kuvassa 3 esitetään kuvitteellinen tilanne, jossa ylempi pylväsdiagrammi esittää riippuvuuden rahankeruuyksiköiden lukumäärän ja yksikön koon suhteen.



Kuva 3. Dynaamiset muutokset visualisoinnissa osatekijöiden perusteella

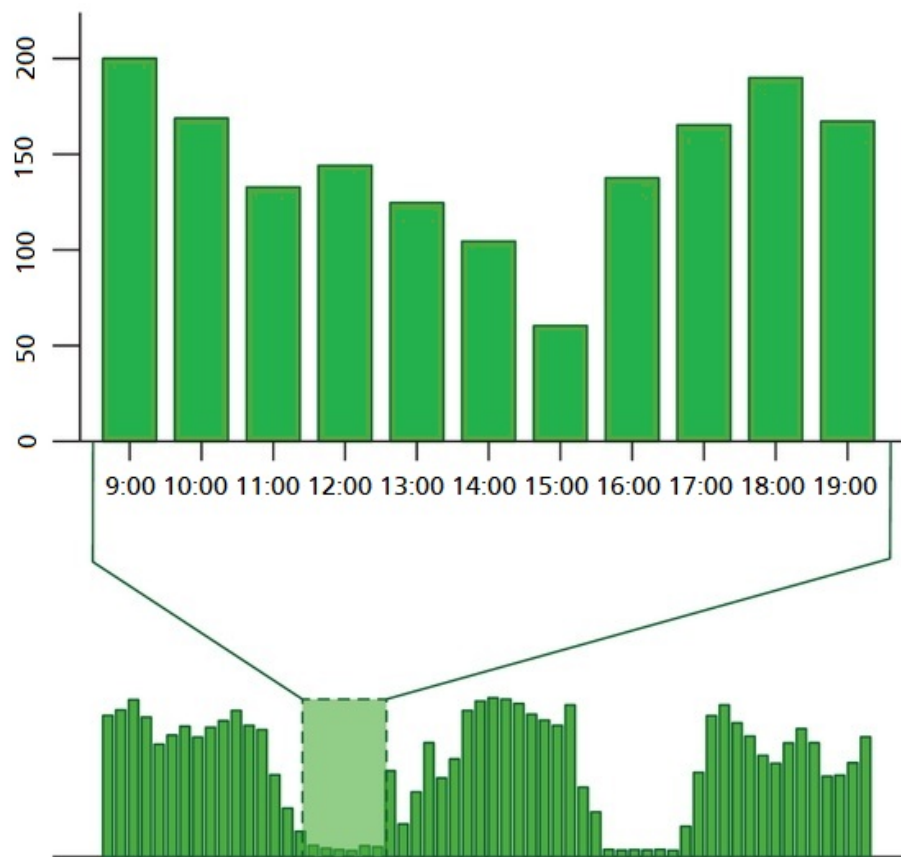
Kun tarkastelija on valinnut toisen osatekijän, esimerkiksi käyttökustannukset, diagrammin tyyppi muuttuu pistediagrammiksi. Kuvan 3 alempi näkymä esittää käyttökustannuksien jakaantumista kunkin rahankeruuyksiköiden välillä. Osatekijöiden lukumäärää tulee voida kasvattaa tai vähentää johdonmukaisesti, jolloin muutokset tulevat näkyviin visualisoinnissa. Prosessi on luonteeltaan iteratiivista, jolloin muutoksia voi tehdä ja toistaa, kunnes saavutetaan haluttu lopputulos [Panopticon Software 2017].

### 3.3 Tietueiden suodattaminen

Eri arvojen selkeä erottuvuus toisistaan on ollut ajankohtainen aihe visuaalisen analyysin saralla, ja aiheen tärkeys on korostunut big datan kontekstissa [SAS Institute 2017]. Vaikka näkymässä olisi samanaikaisesti sata uniikkia arvoa, puhumattakaan sadoista tuhansista arvoista, arvojen nimikkeiden asettaminen on vaikeaa. Lisäksi yhdessä tietojen joukkiossa saattaa olla lukuisia eri arvojoukkoja, joka johtaa tilanteeseen, jossa tietyt arvot hukkuisivat arvojen alle, joilla on korkeammat laajuudet. [Blue 2011.] Näin ollen kokonaisen diagrammin havainnointi voisi olla monimutkaista. Esimerkkinä tilanteesta voisi olla yritys, joka toimii kellon ympäri. Yrityksellä on eri määrä asiakkaita eri kellonaikoina, mutta asiakasmäärien tarkastelu kellonajan mukaisesti vaikeuttaa hilaisten tuntien havainnointia, jolloin asiakasmäärien laajuudet ovat lähes samat ja merkittävästi pienemmät kuin kiireisinä tunteina.

Aineiston tarkastelijat haluavat usein nähdä sekä kokonaiskuvan aineistosta että osakuvauksia yksityiskohtaisemmasta tiedosta, joka saattaa olla tutkimuksen kannalta kiinnostavaa. Lisäksi mielenkiinnon kohde aineistossa ei välttämättä ole staattista, vaan voi muuttua dynaamisesti tutkimusprosessin aikana. [Blue 2011.]

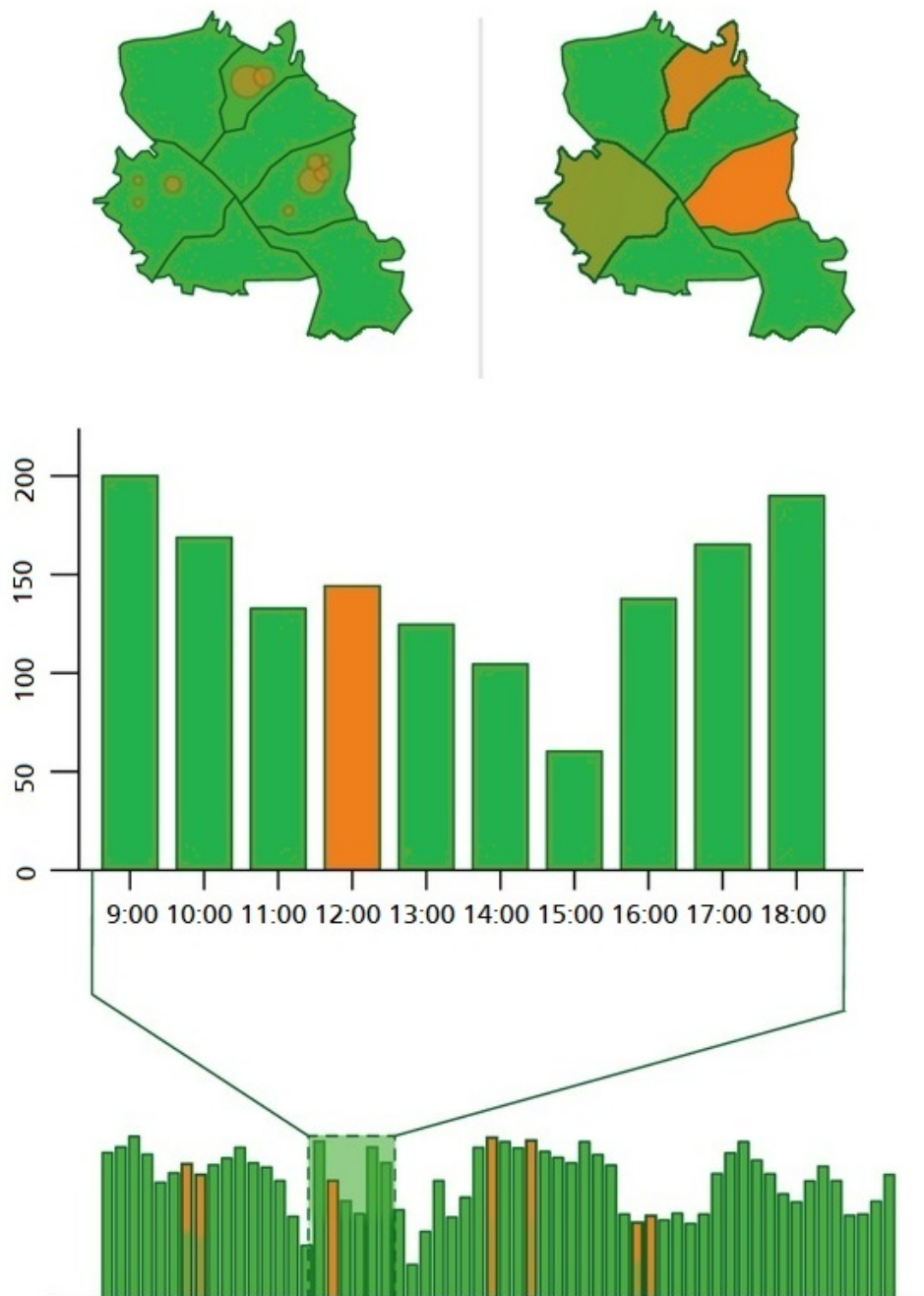
Suodattamisjärjestelmää ja karttaa kokonaisuudesta käytetään lähestymistapana ratkaisemaan ongelma (kuva 4). Aineiston tarkastelija voi muuttaa kartan skaalaa ja nähdä yksityiskohtaisempaa visualisointia tiedoista skaalan sisällä.



Kuva 4. Tarkempi visualisointi kokonaisaineistosta valinnan perusteella

Kuvaustason ei tarvitse olla rajoitettu vain yhteen tarkkuuden tasoon, kuten kuvassa 4, vaan näkymässä voidaan siirtyä tarkemmalle tai laueammalle tasolle jokaisella iteraatiokerralla.

Toisena esimerkkinä tarkastelija voi valita yhden arvojoukon, joka on tarkastelualueen sisällä, nähdä arvojoukon jakautumisen kaupungin kartalla tai korostaa samankaltaisia objekteja kokonaisuutta esittävällä karttakuvalla (kuva 5).



Kuva 5. Tarkempi tiedon visualisointi iteratiivisesti valintojen perusteella

Tietueiden suodattaminen eri kriteerien perusteella on ajankohtainen avain visuaalisen data-analyysin alalla. Ihmisen kognitio ei pysty havaitsemaan suuria joukkoja näkyviä objekteja samanaikaisesti, joten objektien määrän rajoittaminen on luonnollinen vaatimus datan visualisoinnissa. Tietueiden suodattamisen tärkeimmät konseptit ovat esitelty lyhyesti seuraavissa alakohdissa.

Osa toimintamalleista, joita käytetään analyttisessä tutkimusprosessissa, voidaan tunnistaa. Suosituimmat näistä malleista ovat dynaamiset kysely-suodattimet, jotka voidaan ryhmitellä ja yhdistää yksinkertaisempiin käyttöliittymän elementteihin, jolloin käyttöliittymän elementtien avulla voidaan suo-

dattaa tietoa. Suodattimien käyttö avustaa etenkin rutiinitehtävissä, jolloin suodatinelementin painallus rajaa näkymää, jolloin tarkastelijan on helpompaa päätyä haluttuun lopputulokseen tarkastellessaan informaatiota. [Ahlberg and Shneiderman 1994.]

Sirontanäkymän, eli niin kutsutun starfield displayn, lähtökohtaolettaus on, että koko aineisto on jatkuvasti nähtävillä. Korkeimmalla tasolla data vaatii yhdistämistä, jolloin tarkastelija näkee kerralla yhteen liitettyä tietoa, mutta tehdessään tarkempia kyselyitä zoomaamalla lähemmäs tarkasteltavaa kohdetta, kohde purkaantuu yksityiskohtaisemmaksi tiedoksi. [Selassie *et al.* 2011.] Sirontanäkymä muistuttaa siis tähtikarttaa, jossa näkymää voi siirtää lähemmäs objekteja tai kauemmas objekteista.

Joidenkin käyttöliittymäelementtien toiminta voidaan suoraan liittää yhteen toistensa kanssa, jolloin liitosten teko estää käyttäjää tekemästä virheitä syötettäessä dataa tai tarkasteltaessa sitä. Yksinkertaisin esimerkki tiivistä liitoksista, niin sanottu tight coupling, on radiopainikkeet, jotka on liitetty toisiinsa. Yhden radiopainikkeen painallus saa aikaan toiminnan, jossa toisen radiopainikkeen valinta häviää. Monet suodattimet, jotka perustuvat käänteisiin valintoihin, on toteutettu käyttäen tiiviiden liitosten toimintamallia. [Selassie *et al.* 2011.]

## **4 Big datan visualisointimetodit**

Tutkielman viimeinen luku keskittyy konkreettisiin big datan visualisointim metodeihin. Kunkin metodin omassa kohdassa on kuvailtu lyhyesti käytettävää visualisointimethodia ja esitetty arvio metodin vahvuuksista ja heikkouksista. Metodit liittyvät aikaisemmin esitettyihin big datan luokituksiin (Volume-Velocity-Variety), joidenka kautta metodeja arvioidaan.

### **4.1 TreeMap**

Treemap on visualisointimethodi, jossa tyhjää tilaa täytetään hierarkkisella datalla, jonka tulee olla metodin määritelmän mukaan hierarkkisesti linkitettyinä toisiinsa [Tennekes and de Jonge 2011]. Methodi näyttää käytännössä suurelta suorakaiteelta, joka on ryhmitelty ja jaoteltu pienemmiksi suorakaiteiksi, jotka edustavat eri tietoja, ja joiden koko vastaa arvojen suuruuksia aineistossa. [Tennekes and de Jonge 2011.] Esimerkkejä treemap-visualisoinnista ovat tietokoneen muistin tai kovalevyn vapaan tilan visualisointi tai eri organisaatioiden ja organisaation yhteistyökumppaneiden kannattavuuskaaviot.



Kuva 6. Esimerkki treemap -visualisointimenetelmästä [Panopticon Software 2017]

Metodia voidaan käyttää hyväksi määrältään suuriin aineistoihin, joissa voidaan tarkastella iteratiivisesti kunkin eri hierarkiatason näkymää. Käyttäjä voi siirtyä tarkastelemaan seuraavaa lohkoa ja siirtyä eri lohkojen sisällä tarkemmille tarkastelutasoille. Koska metodi perustuu muotoihin, jotka arvioivat kunkin tietueen suuruutta, jokainen muutos tarkasteltavassa aineistoissa vaatii hierarkiatason näkymän piirron uudelleen. Vastaavasti tarkasteltavaa hierarkiatasoa korkeammat tasot eivät vaadi näkymän päivitystä datan muuttuessa, sillä tasot eivät ole suoraan käyttäjän nähtävillä ilman paluuta korkeammalle tasolle.

Visualisoinnissa käytetään hyväksi vain kahta eri osatekijää: lohkon kokoa ja lohkon väriä. Lohkon koko edustaa tietueen suuruutta, ja väriä käytetään visualisoinnissa ryhmittelemään suorakaiteita. On myös hyvä huomioida, että treemapin suuruuksien arviointi vaatii käytettäväksi laskettavissa olevia tietotyyppisiä. Big datan luokitusten perusteella treemap ei täytä kriteereitä, joissa datan tyyppien pitäisi olla moninaisia tai että käsittelyn nopeuden tulisi olla nopea, sillä treemapin visualisointi on läpileikkaus aineistosta yhdessä pisteessä ajassa. Lisäksi kaikkien treemapin tietotyyppien tulee olla laskettavissa olevia.

Menetelmän vahvuuksia ovat [Panopticon Software 2017]

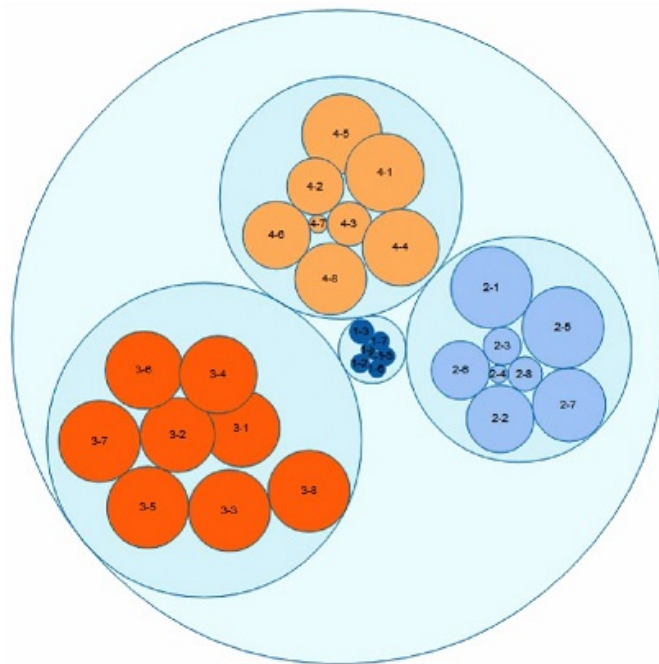
1. Hierarkkinen ryhmittely näyttää selkeästi tietojen suhteet toisiinsa nähden.
2. Poikkeavat havainnot ovat heti nähtävissä huomiovärien avulla.

Menetelmän heikkouksia ovat

1. Datan tulee olla hierarkkista, minkä lisäksi treemap metodina soveltuu paremmin tutkimaan aineistoja, joissa on ainakin yksi tärkeä kvantitatiivinen ulottuvuus.
2. Menetelmä ei sovellu hyvin visualisoimaan historiallisia trendejä tai muutoksia eri aikaväleillä.
3. Lohkojen kokoja laskeva osatekijä ei voi saada negatiivisia arvoja.

## 4.2 Circle packing

Circle packing [Tedesco *et al.* 2013] on suora vaihtoehto treemap -visualisoinnille. Sen lisäksi, että metodissa käytetään muotoina ympyröitä, muotoja voidaan sisällyttää korkeamman hierarkiatason ympyröiden sisälle. Circle packing -visualisoinnin selkeä ansio aikaisemmin esitettyyn treemappiin verrattuna on, että metodin avulla voidaan sekä sisällyttää että havaita enemmän objekteja samassa näkymässä [Tedesco *et al.* 2013].



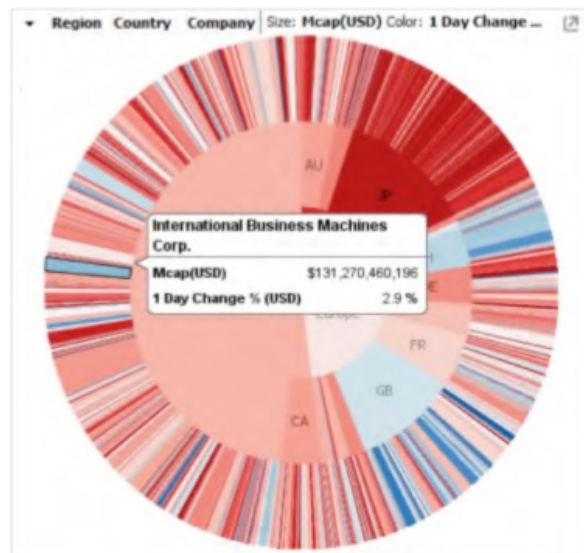
Kuva 7. Circle packing -visualisointimenetelmän esimerkki [Tedesco *et al.* 2013]

Koska metodi perustuu treemap-metodiin, sillä on samat ominaisuudet. Näin ollen voidaan olettaa, että metodi soveltuu big datan kontekstissa suurille tietomäärille. Vaikka metodi on samankaltainen treemapin kanssa, on sen päävahvuus aikaisemmin mainittuun verrattuna tilankäytön tehokkuus näyttöpäätteellä. Circle packing jakaa samat heikkoudet treemapin kanssa.



### 4.3 Sunburst

Sunburst visualisointimetodi on myös vaihtoehto treemapille, vaikka visualisoinnissa käytetään samaa toimintatapaa [Panopticon Software 2017]. Mallissa parametrit on muutettu suuntaan, jossa käytetään kompassimaista näkymää, jonka napojen suuntaan säteet avautuvat. Sunburst-mallissa parametrit eivät ole korkeus ja leveys treemapin tavoin, vaan säteen ja kaaren pituudet. Tämän eroavaisuuden avulla koko diagrammia ei tarvitse päivittää uudelleen datan muuttuessa, sillä uutta dataa sisältävän sektorin suuruutta voidaan muuttaa sektorin säteen avulla. Metodin ominaisuuden perusteella sunburst-visualisoinnilla voidaan näyttää datan dynaamisia muutoksia käyttäen apuna animaatiota.



Kuva 8. Esimerkki sunburst-visualisointimenetelmästä [Panopticon Software 2017]

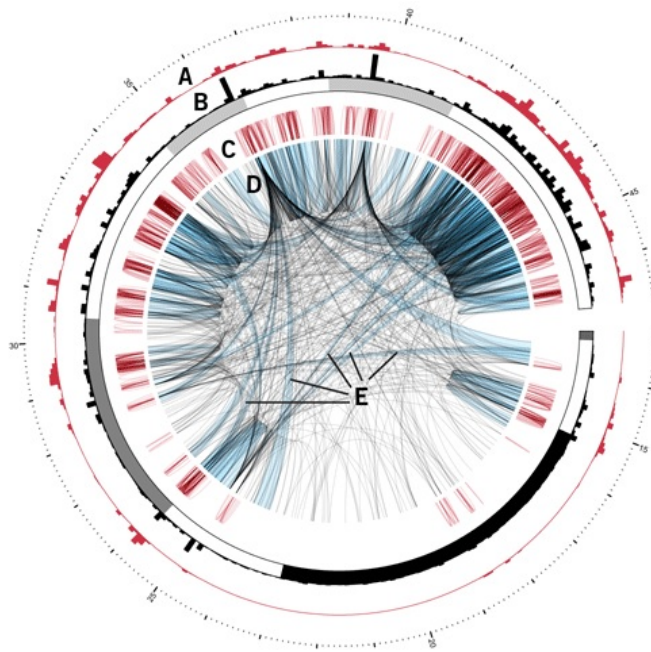
Animaation avulla muutoksia datassa voidaan visualisoida näkymässä dynaamisesti, joten big datan luokituksien perusteella sunburst soveltuu datan visualisointiin, joka vaatii käsittelynopeutta. Kahden aikaisemman metodin tavoin visualisointimetodilla on samat heikkoudet ja vahvuudet kuin treemapilla ja circle packing -metodilla. Lisäksi metodi on helposti havainnoitavissa [Cawthon and Moere 2007].

### 4.4 Circular network diagram

Circular network diagram -visualisointimetodissa eri objektit sijoittuvat ympyrän kehälle, josta objektit liittyvät toisiinsa kaarilla perustuen objektien suhteisiin [Krzywinski *et al.* 2009]. Suhteiden toistuvuutta ja vahvuutta mitataan vi-

sualisoinnissa kaarien paksuudella ja värien kylläisyydellä. Metodi on usein myös vuorovaikutteinen, missä diagrammi piilottaa tarpeettomat linkit objektien välillä ja korostaa valittuja objekteja. Circular network diagram alleviivaa suoria yhteyksiä eri objektien välillä ja näyttää, kuinka vahvoja eri suhteet ovat. [Krzywinski *et al.* 2009.]

Esimerkkitilanteita, joissa diagrammia voidaan käyttää hyödyksi, ovat muun muassa tuotteiden kuljetusmäärät eri kaupunkien välillä tai eri tuotteiden tilausmäärät kauppojen välillä.



Kuva 9. Esimerkki circular network -visualisoinnista [Krzywinski *et al.* 2009]

Metodin avulla voidaan luoda näkymä yhdistetystä tiedosta kaarina eri objektien välillä, jolloin tarkastelija saa selville lukumääräsuhteet objektien välillä. Diagrammia voidaan käyttää hyväksi suurien aineistojen visualisointiin, eri suuruisia objekteja voidaan asettaa diagrammin kehälle, ja kaarien kokoa objektien välillä voidaan muuttaa. Lisäksi diagrammiin voidaan lisätä ylimääräistä tietoa eri objektien arvoista, jolloin diagrammi ei rajoita eri tietotyyppien sijoittamista samaan näkymään ja tekemästä yhteyksiä objektien välillä. Diagrammista voi tulla vaikeasti ymmärrettävä, mutta joissakin tapauksissa visualisointimetodi tarjoaa riittävästi informaatiota, jotta metodi on hyödyllinen tarkastelijalleen ja päätöksenteolle. Circular network diagram soveltuu esittämään monimuotoisia tietotyypppejä.

Diagrammin pyöreä muoto auttaa lisäksi diagrammin luettavuudessa, sillä katse seuraa luontaisesti pyöreiden kaarien linjoja suorakaiteiden siksak-liikkeen sijaan [Krzywinski *et al.* 2009]. Kuitenkin diagrammin kokonaisnäkymän vuoksi jokainen muutos aineistossa vaatii diagrammin piirron uudelleen.

Menetelmän vahvuuksia ovat [Krzywinski *et al.* 2009]

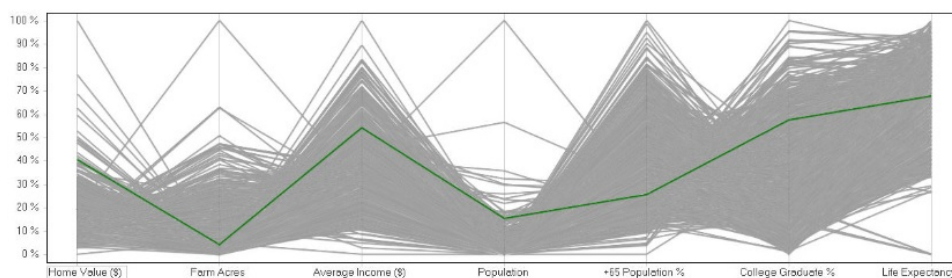
1. Diagrammin avulla voidaan tehdä suhteellisia esityksiä aineistosta, jota on helppo havainnoida.
2. Ympyrän sisällä tarkkuustaso vaihtuu tasaisesti säteittäin. Tästä johtuen ympyrän sisällä voi esittää tiiviisti yhteenvetoja tai osoittaa kiinnostavia kohteita.

Menetelmän heikkouksia ovat [Krzywinski *et al.* 2009]

1. Visualisointi saattaa johtaa lähes huomaamattomaan esitysmuotoon, joka saattaa vaatia eri objektien uudelleenjärjestelyä näyttöpäätteellä.
2. Pienimmät objektit saattavat jäädä parametrillisesti suurempien objektien varjoon, jolloin informaatiosta ei ole tutkijalle mitään hyötyä [Hoek 2011].

#### 4.5 Parallel coordinates

Parallel coordinates -visualisointimetodin avulla voidaan analysoida useita tiedon osatekijöitä eri objekteista [Few 2006]. Kaikki tutkittava data asetetaan diagrammissa yhdelle akselille, jonka jälkeen objektia vastaavat tietojen arvot suhteellisella mittakaavalla asetetaan toiselle akselille. Jokaista tietoa sisältävää objektia edustaa sarja yhteen liitettyjä poikittaisviivoja, jotka näyttävät objektin paikan verrattuna muihin objekteihin. Metodin avulla voidaan visualisoinnissa käyttää paksua viivaa näyttöpäätteellä kuvaamaan yksittäisiä objekteja, jolloin diagrammilla voidaan osoittaa suuria tietomääriä. [Few 2006.]



Kuva 10. Parallel coordinates -visualisointimenetelmän esimerkkikuva [Few 2006]

Tavallista kaksiulotteista rinnakkaisten koordinaattien diagrammia voidaan jatkaa kolmiulotteiseksi, monisuhteiseksi diagrammiksi. Kolmiulotteisessa mallissa akselit sijoittuvat tasaisesti ympyrän keskipisteen ympärille. Tietoa sisältävä objekti näytetään mallissa jaoteltuna segmentteihin, jotka läpäisevät kaikki kolme akselia. Kolmiulotteisen mallin etuna on, että kaikkia parittaisia suhteita mallin keskustan muuttujan ja mallin uloimpien muuttujien välillä voidaan tarkastella samanaikaisesti. [Johansson *et al.* 2008.]



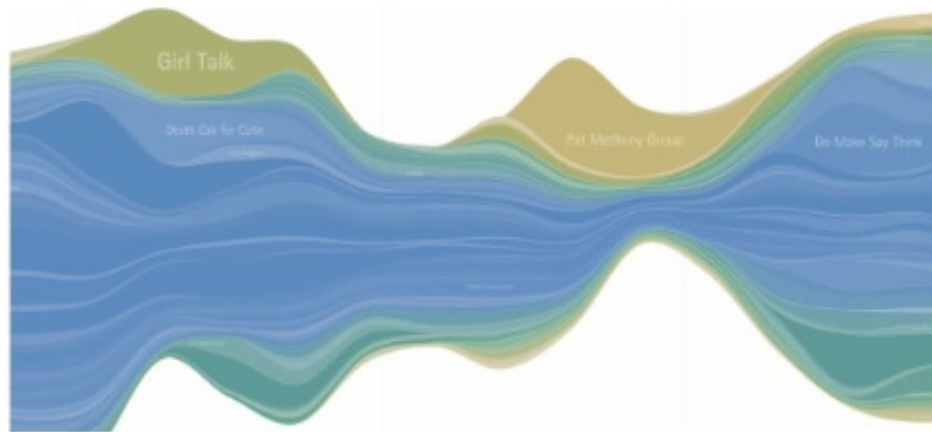
Kuva 11. Kolmiulotteinen parallel coordinates -visualisointi [Johansson et al. 2008]

Menetelmän avulla voidaan tarkastella suuria määriä eri objekteja samalla näyttöpäätteellä, joten diagrammi soveltuu esittämään monimuotoisia tietotyyppisiä. Koska visualisointimetodi perustuu suhteellisiin arvoihin, jokaiselle osatekijälle tulee laskea minimi- ja maksimi-arvot. Mikäli kunkin objektin minimi- ja maksimi-arvot pysyttelevät diagrammin suurimpien ja pienimpien arvojen välissä, diagrammia ei tarvitse piirtää kokonaan uudelleen, jos arvot muuttuvat dynaamisesti, joten metodin avulla voidaan visualisoida dynaamista dataa. [Johansson *et al.* 2008.]

Ongelmallista tämän visualisointimenetelmän käytön yhteydessä on se, että samanaikaisesti esitettävien osatekijöiden määrällä on rajoitteita. Lisäksi dynaamiset muutokset datan visualisoinnissa voivat muuttaa kokonaisnäkymää, jos yksittäisten objektien arvot ylittävät tai alittavat näkymän huippu- ja minimiarvot. Selkeä vahvuus menetelmässä on, että eri osatekijöiden järjestely diagrammissa ei vaikuta kokonaisnäkymän havaitsemiseen. Metodin avulla voidaan myös tarkastella ja analysoida kokonaista joukkiota tietoa sisältäviä objekteja samanaikaisesti sekä keskittyä yksittäisiin objekteihin. [Johansson *et al.* 2008; Edsall 1999.]

#### 4.6 Streamgraph

Viimeisenä esiteltävänä visualisointimetodina on niin kutsuttu streamgraph, joka on diagrammi, jossa on toistensa päälle pinottuja alueita, jotka sijoittuvat keskiakselin ympärille [Byron and Wattenberg 2008]. Diagrammi näyttää soljuvalta elolliselta muodolta, josta nimitys tulee. Metodin avulla voidaan näyttää eri tapahtumaketjujen muutoksia, esiintymien lukumääriä, sekä tapahtumaketjun tilaa verrattuna muuhun tietoon. Streamgraph voi sisältää sarjan samankaltaisia tapahtumia, jotka esitetään aikajanalla näkymässä. [Byron and Wattenberg 2008.]



Kuva 12. Streamgraph -visualisointimenetelmän esimerkki [Byron and Wattenberg 2008]

Diagrammin avulla voidaan esittää käytännössä muun muassa musiikkiin liittyviä trendejä tai elokuvaan liittyviä trendejä.

Menetelmällä on kaksi toisiinsa liittyvää päämäärää: näyttää useita yksittäisiä aikajanoja ja välittää tarkastelijalle aikajanojen summa. Koska yksittäisten alueiden korkeudet asettuvat päällekkäin kokonaisnäkymään, voidaan diagrammin avulla ilmaista molemmat päämäärät samanaikaisesti. Tästä aiheutuu kuitenkin se, että alueiden välillä ei voi olla tyhjää tilaa, sillä tyhjä tila vääristäisi alueiden summaa. Mikäli eri aikajanojen väliin jääviin alueisiin tulee muutoksia dynaamisesti, koko diagrammi värähtelee pystyakselin suuntaisesti, vaikka muihin aikajanoihin ei tulisikaan muutoksia. [Byron and Wattenberg 2008.]

Streamgraph soveltuu käytettäväksi vain yhden tiedon ulottuvuuden mittaamiseen, eikä diagrammi tue big datan monia tiedon eri formaatteja, mutta diagrammia voidaan silti käyttää hyväksi suurien tietomäärien tarkasteluun. Diagrammia voidaan jatkaa lisäämällä siihen dynaamisesti uusia arvoja, mutta diagrammin avulla ei voida esittää muuta tietoa kuin kvantitatiivisia osatekijöitä ja tekijöiden muodostamaa kokonaisuutta. Menetelmän avulla voidaan tehokkaasti visualisoida trendejä ja niiden muutoksia aikajanalla, mutta metodin heikkouksia ovat se, että näkymä aineistosta esittää vain yhtä tiedon osatekijää, ja että metodi on riippuvainen eri kerrosten eli objektien lajittelusta diagrammiin. [Byron and Wattenberg 2008; Lee *et al.* 2010.]

## 5 Yhteenveto

Suurien tietomäärien aikakaudella perinteiset informaation visualisointimenetelmät törmäävät rajoitteisiin, jotka juontuvat haasteista, joita big data asettaa toimivalle visualisoinnille ominaispiirteidensä vuoksi (Volume-Velocity-Varie-

ty). Tutkielmassa esiteltiin merkittävimpiä ongelmia big datan visualisoinnissa ja tarkasteltiin lähtökohtia, joiden avulla ongelmia voidaan välttää visualisoitaessa tietokoneavusteisesti suuria aineistomääriä.

Tutkielma esitteli myös kuusi vakiintunutta visualisointimenetelmää tarkemmin ja arvioi menetelmiä big datan käyttökontekstissa. On kuitenkin hyvä huomioida, että menetelmiä on lukuisia: Panopticon Softwaren mietintö (niin kutsuttu white paper) esittelee lyhyesti yli kaksikymmentä, joista kuitenkin valtaosa on toimintamalleiltaan hyvin samankaltaisia jo olemassa olevien menetelmien kanssa. Useimmissa tapauksissa esitetty visualisointimenetelmä rakentuu jo olemassa olevan menetelmän varaan muuttaen aikaisempaa visualisointimenetelmää vastaamaan paremmin uutta käyttötilannetta. Esimerkiksi tutkielmassa arvioitu sunburst-visualisointimenetelmä rakentuu perinteisen ympyräkaavion varaan.

Menetelmien soveltuvuutta big datan visualisointiin voisi tutkia kriittisesti lisää ja muodostaa tutkimuksien perusteella suosituksia eri menetelmien toimivuudesta eri käyttötilanteissa. Suuret aineistot syntyvät lukuisten tahojen yhteistoiminnan tuloksena, tutkijat ja ohjelmistoteollisuus kehittävät jatkuvasti uusia työkaluja visualisointitehtäviin, joten tutkielman kirjoittamisen aikaan seuraava mielenkiintoinen suunta tutkimukselle voisi olla yhteistoiminnallinen (collaborative) data-analyysi eri visualisointityökalujen avulla, jotka tukevat samanaikaista työskentelyä aineistojen parissa.

## Viiteluettelo

- Agrawal, R., Kadadi, A., Dai, X., & Andres, F. (2015). Challenges and opportunities with big data visualization. In: *Proc. of the 7th International Conference on Management of Computational and Collective intelligence in Digital Ecosystems*, 169-173.
- Ahlberg, C., & Shneiderman, B. (1994). Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In: *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*. 313-317.
- Blue, D. (2011). Big Data: Big Opportunities to Create Business Value. <http://www.emc.com/microsites/cio/articles/big-data-big-opportunities/LCIA-BigData-Opportunities-Value.pdf>. Checked 14.11.2017.
- Byron, L., & Wattenberg, M. (2008). Stacked graphs - geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics* 14(6), 1245-1252.
- Card, S. K., Mackinlay, J. D., & Shneiderman, B. (1999). *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann.



- Cawthon, N., & Moere, A. V. (2007). The effect of aesthetic on the usability of data visualization. In: *Proc. of the 11th International Conference on Information Visualization*.
- Chen, C. (2006). *Information Visualization: Beyond the Horizon*. Springer.
- Cleveland, W. S., & McGill, R. (1984). Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association* 79(387), 531-554.
- Dijks, J. -P. (2013). Big Data for the Enterprise: Oracle White Paper. Oracle Published Group. <http://www.oracle.com/us/products/database/big-data-for-enterprise-519135.pdf>. Checked 18.11.2017.
- Edsall, R. (1999). The dynamic parallel coordinate plot: visualizing multivariate geographic data. In: *Proc. of the 19th International Cartographic Association Conference*.
- Few, S. (2006). Multivariate Analysis Using Parallel Coordinates. [http://www.perceptualedge.com/articles/b-eye/parallel\\_coordinates.pdf](http://www.perceptualedge.com/articles/b-eye/parallel_coordinates.pdf). Checked 2.12.2017.
- Heer, J., & Shneiderman, B. (2012). Interactive dynamics for visual analysis. *Queue – Microprocessors* 10, 2.
- Hoek, P. (2011). Parallel arc diagrams: Visualizing temporal interactions. *Journal of Social Structure* 12(7).
- Johansson, J., Forsell, C., Lind, M., Cooper, M., (2008). Perceiving patterns in parallel coordinates: Determining thresholds for identification of relationships. *Information Visualization* 7(2), 152-162.
- Kirk, A. (2012). *Data Visualization: A Successful Design Process*. Packt Pub.
- Krzywinski, M., Schein, J., Birol, I., Connors, J., Gascoyne, R., Horsman, D., . . . Marra, M. A. (2009). Circos: An information aesthetic for comparative genomics. *Genome Research* 19(9), 1639-1645.
- Lee, T. -Y., Jones, C., Chen, B. -Y. and Ma, K. -L. (2010). Visualizing data trend and relation for exploring knowledge. In: *Proc. of the IEEE Pacific Visualization Poster*.
- Panopticon Software. (2017). Understanding the value of visual data discovery – A guide to visualizations. <http://www.panopticon.com/resource-center/literature/visualizations-guide/>. Checked 3.12.2017.
- Robertson, G., Fernandez, R., Fisher, D., Lee, B., & Stasko, J. (2008). Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics* 14(6), 1325-1332.
- SAS Institute. (2017). Data Visualization Techniques. [https://www.sas.com/content/dam/SAS/en\\_us/doc/whitepaper1/data-visualization-techniques-106006.pdf](https://www.sas.com/content/dam/SAS/en_us/doc/whitepaper1/data-visualization-techniques-106006.pdf). Checked 4.12.2017.

- Selassie, D., Heller, B., & Heer, J. (2011). Divided edge bundling for directional network data. *IEEE Transactions on Visualization and Computer Graphics* 17(12), 2354-2363.
- Tedesco, J., Dudko, R., Sharma, A., Farivar, R., & Campbell, R. (2013). Theius: A streaming visualization suite for hadoop clusters. In: *Proc. of the 2013 IEEE International Conference on Cloud Engineering (IC2E)*, 177-182.
- Tennekes, M., & De Jonge, E. (2011). Top-down data analysis with treemaps. In: *Proc. of IMAGAPP/IVAPP*, 2011.
- Zikopoulos, P. (2012). *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*. McGraw-Hill.



# Tekstinsyöttö laajennetussa todellisuudessa

**Petja Makkonen**

## **Tiivistelmä.**

Tässä tutkielmassa tutustun joihinkin laajennetun todellisuuden laitteisiin erityisesti niiden tekstinsyötön kannalta. Vertailen näiden laitteiden käytössä olevia ja tulevaisuuden tekstinsyöttötapoja keskenään. Näin muodostuu kuva siitä, mitä tulee ottaa huomioon suunniteltaessa tapoja käyttäjän tekstisyöttöön laajennetuissa todellisuuksissa.

**Avainsanat ja -sanonnat:** Laajennettu todellisuus, virtuaalitodellisuus, sekoitettu todellisuus, tekstinsyöttö, käytettävyys.

## **1. Johdanto**

Tässä tutkielmassa tutustutaan laajennetun todellisuuslaitteiden ratkaisuihin käyttäjän tekstinsyötön kannalta. Laajennettu todellisuus voidaan jakaa alakäsitteisiin virtuaalitodellisuus, sekoitettu todellisuus ja lisätty todellisuus. Tutkielmaan on valittu kolme laitetta kahdelta eri osa-alueelta.

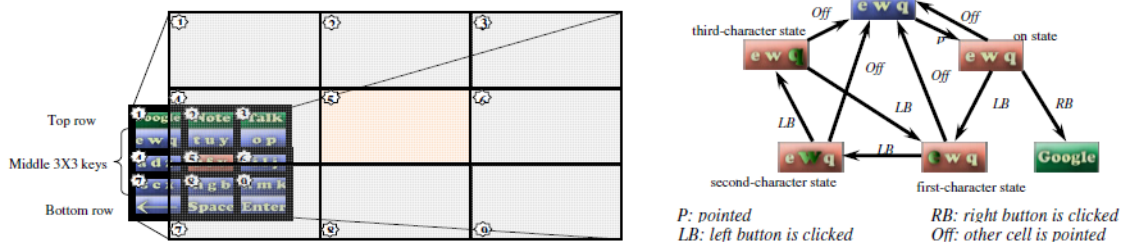
Käyttäjän tekstinsyötöllä tarkoitetaan tapoja ja käytäntöjä, joilla käyttäjä saa syötettyä järjestelmään toivomaansa syötettä tekstimuodossa. Tutkielmassa tutustutaan laitteiden vakiokäytäntöihin tekstinsyötössä ja tarjolla oleviin tunnetuimpiin vaihtoehtoihin tapoihin syöttää tekstiä järjestelmään. Käytäntöjä vertaillaan, jotta saataisiin selville, mitä tekstinsyötön kannalta tulee laajennetussa todellisuudessa ottaa huomioon. Lopuksi pohditaan alan tulevaisuutta.

Sekä virtuaalitodellisuus että sekoitettu todellisuus ovat kasvavia teknologia-aloja [Statista 2017]. Kyseisiä teknologioita otetaan jatkuvasti entistä laajamittaisempaan käyttöön niin viihteessä kuin teollisuudessaakin. Koska virtuaalitodellisuus ja sekoitettu todellisuus ovat suhteellisen uusia teknologioita ei ratkaisut tekstinsyötössä esimerkiksi käyttäjäkokemuksen suhteen ole vielä vakiintuneet optimaaliselle tasolle. Tästä syystä aihetta on hyvä tutkia.

## **2. Aiemmat tutkimukset**

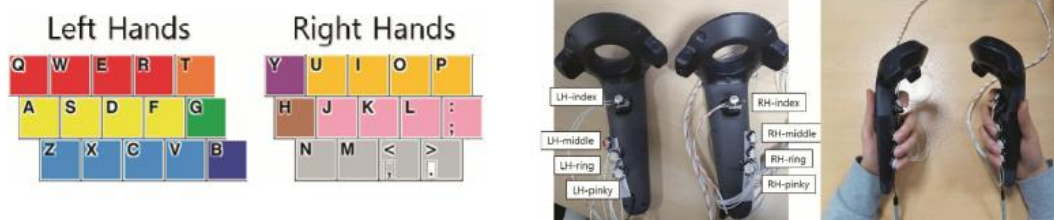
Min [2011] esittää ratkaisuksi virtuaalitodellisuuden tekstinsyöttöön 3x3-koosta taulukkoa, jossa jokainen taulu sisältää joukon kirjaimia. Ratkaisu on esitetty kuvassa 1. Ratkaisussa jäljitellään matkapuhelinten tekstinsyöttöä ja sen

tarkoituksena on parantaa tekstinsyötön helppoutta ja tarkkuutta, sillä yksittäisen kirjaimen valinta virtuaaliselta QWERTY-näppäimistöltä on hankalaa virtuaalitodellisuuslaitteen osoittimilla. Tässä taulukkomallissa käyttäjä osoittaa haluamaansa taulukon osaa, joka sisältää käyttäjän tavoittelemat kirjaimet. Taulukon osa korostuu ja käyttäjä voi fyysisillä näppäimillä korostaa haluamansa kirjaimen taulukon osan sisältä. Kun käyttäjä on korostanut haluamansa kirjaimen, valitsee hän sen käytettäväksi toisella fyysisellä näppäimellä. Verrattuna normaaliin virtuaaliseen QWERTY-näppäimistöön pärjäsi Minin ratkaisu sitä paremmin. [Min 2011]



Kuva 1. Minin [2011] ratkaisu

Lee ja Kim [2017] kehittivät vaihtoehtoisen tavan tekstinsyöttöön virtuaalitodellisuuslaitteiden puutteellisen sormien jäljittämisen takia. Ratkaisu on esitetty kuvassa 2. Ratkaisussa HTC Viven ohjainten taakse kiinnitettiin näppäimet jokaiselle sormelle tekstinsyötön helpottamiseksi. Käyttäjä valitsi haluamiaan kirjaimia ylimääräisillä näppäimillä virtuaalisesta QWERTY-näppäimistöstä. Tutkimuksessa tämän vaihtoehtoisen tavan todettiin olevan perinteistä tekstinsyöttömenetelmää parempi.



Kuva 2. Leen ja Kimin [2017] ratkaisu

Aiheesta on tehty paljon tutkimuksia, joista suurin osa keskittyy ulkoisen laitteen apuun. Näitä ratkaisuja ovat esimerkiksi ”käsinkirjoitus” ohjaimen avulla tai kirjoittamiseen tarkoitettu hansikas.

### 3. Virtuaalitodellisuus

*Virtuaalitodellisuudella* (virtual reality) tarkoitetaan todellisuutta, jonka ihminen on luonut tietokoneen ja sen lisälaitteiden avulla. Ihminen muodostaa kuvan todellisuudestaan aistielinten avulla. Korvaamalla näiden aistielinten ärsykkeet tietokoneella luoduilla ärsykkeillä voidaan luoda todellisuus, jota käyttäjä itse hallitsee. [Bates and Brkljac 2012]. Virtuaalitodellisuus eroaa muista laajennetun todellisuuden aloista siten, että se luo kaikista immersiiivisimmän kokemuksen.

#### 3.1. HTC Vive

HTC Vive -järjestelmä koostuu virtuaalitodellisuus laseista, kahdesta majakasta (lighthouse) sekä kahdesta ohjaimesta [Vive 2017]. Viveä käytetään yhdistämällä se tietokoneeseen. Kokonaisuus on esitetty kuvassa 3. Vive perustuu SteamVR-teknologiaan, jonka avulla voidaan määrittää käyttäjän sijainti ja asento. Majakat lähettävät ympärilleen valonsäteitä, jotka määrittävät ympärillä olevaa tilaa sekä kommunikoivat Viven virtuaalilasien ja ohjainten kanssa. [PC gamer 2015]. Virtuaalitodellisuuden hallintaan käytetään HTC:n ja Valven yhteistyössä kehitettyjä ohjaimia. Ohjaimilla käyttäjän on tarkoitus osoittaa ja kohdentaa virtuaalitodellisuuden elementtejä. Ohjaimet koostuvat kaksitasoisista liipasinnäppäimistä, jotka on sijoitettu etusormien alle, kotinäppäimestä sekä peukalolla hallittavasta kosketuspinnasta. Lisäksi ohjain tukee tuntoaistiin perustuvaa palautetta. [Prasuethsut 2016]. Tekstinsyöttö perustuu ohjaimella tehtäviin toimintoihin.



Kuva 3. HTC Vive [2017]

---

### 3.2. HTC Viven vakiotekstinsyöttö

Koska Viven ohjaus perustuu pääosin mukana tuleviin ohjaimiin, on tekstinsyöttö ratkaistu virtuaalisella näppäimistöllä (virtual keyboard). Virtuaalinen näppäimistö on yksinkertaisimmillaan perinteinen näppäimistö, joka on luotu virtuaalisesti näytölle, josta sitä voi ohjata hiirellä tai vaihtoehtoisella osoittimella.

Kuvassa 4 on esitetty HTC Viven virtuaalinen näppäimistö. Viven virtuaalisen näppäimistön ratkaisu muistuttaa perinteistä virtuaalista näppäimistöä. Näppäimet ovat aseteltu pohjaan QWERTY-näppäimistön tapaan. Valitsemalla näppäimistön tyhjän merkin (blank), saa käyttäjä erikoismerkit esille. Tämä toimii samaan tapaan kuin mobiililaitteista tutuista virtuaalisista näppäimistöistä. Näppäimistöstä löytyy siis valmius kirjoittaa käytännössä mitä tahansa syötettä. Lisäksi näppäimistön voi halkaista keskeltä erottaen näppäimistön puoliskot toisistaan.

Viven näppäimistö sisältää ennakoivan tekstinsyötön, joka löytyy näppäimistön yläreunasta. Lisäksi käyttäjällä on mahdollisuus liittää kopioimaansa tekstiä erillisellä näppäimellä näppäimistön vasemmasta yläreunasta.

Itse tekstinsyöttö näppäimistöllä voidaan suorittaa kahdella tavalla. Toinen vaihtoehtoista on käyttää Viven ohjaimia osoittimina. Tämä tapahtuu siten, että käyttäjä osoittaa haluamaansa kirjainta näkyvissä olevalta virtuaaliselta näppäimistöltä ohjaimella ja valitsee sen valintanäppäimellä. Ohjain toimii ikään kuin laserosoittimeksi, jolla käyttäjä tähtää haluamaansa kirjaimeseen. Osoittamisen avuksi virtuaalisella näppäimistöllä näkyy se kohta, johon ohjain



Kuva 4. HTC Viven virtuaalinen näppäimistö [Yu 2017]

sillä hetkellä osoittaa. Toinen vaihtoehtoisista syöttötavoista on käyttää ohjainten kosketuspintoja. Liikuttamalla peukaloaan kosketuspinnalla osoitin liikkuu näppäimistöllä. Haluamansa kirjaimen kohdalla käyttäjä käyttää taas valintanäppäintä valitakseen kirjaimen. [Yu 2017].

Ratkaisut sisältävät käytettävyyssongelmia. Yun [2017] mukaan molemmat ratkaisusta tuntuvat epäluonnollisilta. Laserosoitinratkaisussa käyttäjä joutuu tekemään paljon työtä sekä käsillään että silmillään. Isolta näppäimistöltä osoittaminen vaatii laajoja käden liikkeitä liikkuesssa näppäimestä toiseen. Lisäksi käyttäjä joutuu seuraamaan osoittimen liikettä katseellaan ja suurikokoisesta näppäimistöstä johtuen on silmien liikeradat huomattavan suuret. Pitkässä käytössä vaativat liikeradat käsillä sekä silmillä voivat aiheuttaa raskautta sekä kärsimättömyyttä käyttäjässä.

Tekstinsyöttö kosketuspinnoin tapahtuu halkaistulla näppäimistöllä. Verrattuna laserosoitinratkaisuun käyttäjän voimat säästyvät, sillä käyttäjän ei tarvitse suorittaa vaativia liikkeitä ohjaimilla. Halkaistun näppäimistön vasenta puolta hallitaan vasemmanpuoleisella ohjaimella sekä näppäimistön oikeaa puolta oikeanpuoleisella ohjaimella. Yun [2017] mukaan halkaistulla näppäimistöllä syöttäminen voi tuntua käyttäjälle epäluontevalle, sillä käyttäjät joutuvat harvoin tilanteisiin, joissa käytetään halkaistua näppäimistöä. Lisäksi käyttäjä voi hallita halkaistun näppäimistön puoliskosta vain kyseisen puoliskon sisältämiä kirjaimia sille tarkoitettulla ohjaimella, eikä siis yhdellä ohjaimella syöttäminen ole käytännössä mahdollista.

Yu lisää, että käytettävyyssongelmat ovat osaksi tottumiskysymyksiä ja harjoittelulla tekstinsyöttäminen voi ruveta tuntumaan luonnollisemmalta.

### **3.3. Sony PlayStation VR**

Sony PlayStation VR -järjestelmä koostuu virtuaalitodellisuuslaseista sekä kamerasta. Kamera sekä virtuaalitodellisuuteen tarkoitetut ohjaimet myydään erikseen ja ne ovat tärkeä osa VR-kokemusta, joten ne lasketaan osaksi järjestelmää. Järjestelmä on esitetty kuvassa 5. Toisin kuin HTC Vive, järjestelmä määrittää käyttäjän sijaintia, ympäristöä ja asentoa kameran avulla, joka seuraa niin laseista sekä ohjaimesta tai ohjaimista lähtevää valoa. [Griffiths 2016]. Ohjatakseen virtuaalitodellisuutta käyttäjä voi käyttää PlayStation 4 -konsolin mukana tulevaa ohjainta tai erikseen myytäviä, virtuaalitodellisuutta varten suunniteltuja, PlayStation Move -ohjaimia. PlayStation Move -ohjaimet ovat perusidealtaan samankaltaiset kuin HTC Viven ohjaimet. Käyttäjä ohjaa virtuaalitodellisuutta osoittamalla ja kohdentamalla sauvojen kärkipistettä virtuaalitodellisuuden elementteihin. Ohjaimiin on myös sisällytetty fyysisiä näppäimiä hallinnan helpottamiseksi. [PlayStation 2017].



Kuva 5. Sony PlayStation VR [PlayStation 2017]

---

### 3.4. Sony PlayStation VR:n vakiotekstinsyöttö

Kuten HTC Vivessä, on myös PlayStation VR:n näppäimistön puute korvattu virtuaalisella näppäimistöllä. PlayStationin virtuaalinen näppäimistö muistuttaa myös perinteistä virtuaalista näppäimistöä, kuten myös Viven tapauksessa. Suuria eroja näiden kahden virtuaalisen näppäimistön toteutuksesta ei löydy, vaan peruselementit ovat samat.

Näppäimistö on muodostettu QWERTY-mallin mukaisesti, joten se on pääpiirteittäin tuttu kaikille käyttäjille. Ratkaisu on kuvassa 6. Kuten Vivessä, näppäimistön yläreunasta löytyy toteutus ennakoivalle tekstinsyöttölle ja näppäimistö sisältää näppäimen, jolla käyttäjä voi liittää kopioimaansa tekstiä. Toisin kuin Vivessä, PlayStation VR:stä löytyy mahdollisuus liikuttaa osoitinta käyttäen virtuaalisesta näppäimistöstä löytyviä sille erikseen tarkoitettuja näppäimiä. [PlayStation 2017]. PlayStationin ratkaisu eroaa Vivestä myös siten, että virtuaalista näppäimistöä ei voi halkaista kahtia.



Kuva 6. PlayStation VR:n virtuaalinen näppäimistö [PlayStation 2017]

---

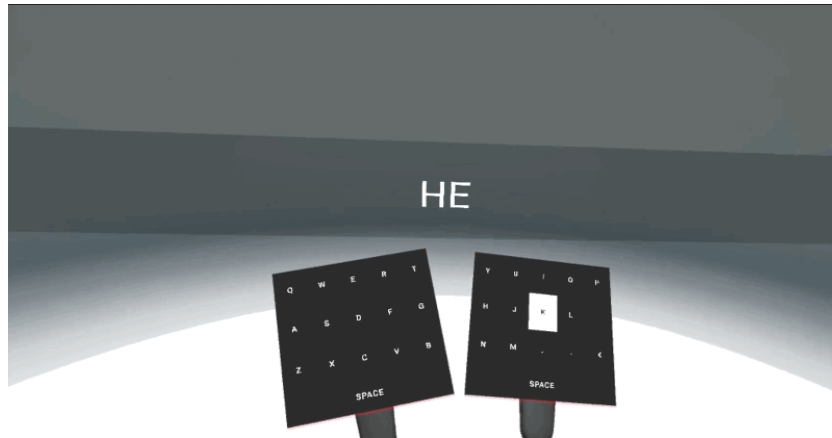
Järjestelmällä on mahdollista syöttää tekstiä peräti kolmella vaihtoehtoisella tavalla [PlayStation 2017]. Ensimmäinen vaihtoehtoista on konsolin mukana tulevalla ohjaimella syöttö. Mukana tuleva konsoliohjain sisältää niin nuolinäppäimet, kosketuspinnan kuin myös liikkumiseen tarkoitetut analogiset sauvat. Syöttäminen on mahdollista sekä nuolinäppäimillä, kosketuspinnalla että sauvoilla. Näppäimistöllä näkyvä osoitin liikkuu kirjaimesta kirjaimeen käyttäjän ohjatessa sitä valitsemallaan tavalla. Kun käyttäjä on haluamassaan kirjaimessa, valitsee hän sen valintanäppäimellä. Nämä tavat muistuttavat hie-  
man Viven kosketuspinta ratkaisua. Molemmissa syöttäminen tapahtuu ainoas-  
taan sormilla.

Ohjainta on mahdollista käyttää myös osoittimena Viven ohjainten tapaan. Asettamalla ohjaimen osoitintilaan, voi käyttäjä syöttää tekstiä osoittamalla ohjaimella haluamaansa kirjaimeen [PlayStation 2017]. Osoitin liikkuu ruudulla näkyvissä olevalla virtuaalisella näppäimistöllä ja halutun kirjaimen kohdalla valitaan kirjain valintanäppäimellä. Kuten Vivessä, ohjain toimii ikään kuin laserosoittimena, jonka osoittama piste näkyy esillä olevalla näppäimistöllä.

Kuten mainittu, kokonaiseen virtuaalitodellisuus kokonaisuuteen kuuluvat PlayStation Move -ohjaimet. Move -ohjaimilla käyttäjä voi syöttää tekstiä kuten Viven ohjaimilla. Ohjaimet toimivat taas ikään kuin laserosoittimina, joilla täh-  
dätään näkyvissä olevaan virtuaaliseen näppäimistöön. Haluamansa kirjaimen  
kohdalla käyttäjä valitsee sen valintanäppäimellä. Käyttäjän osoittama kohta  
näkyy näppäimistöllä.

### **3.5. Vaihtoehtoiset tekstinsyöttötavat**

Ng [2017] on kehittänyt HTC Vivelle vaihtoehtoisen tekstinsyöttötavan, joka on esitetty kuvassa 7. Ratkaisu muistuttaa vakiotekstinsyötön ratkaisua, jossa käy-  
tetään halkaistua näppäimistöä, jota hallitaan ohjaimien kosketuspinoilla. Käyttäjä ohjaa molempia näppäimistön puoliskoja sille tarkoitetulla ohjaimella. Ratkaisussa pyritään tukemaan mobiilikäytöstä opittua lihasmuistia ja tuke-  
maan tätä visuaalisella palautteella tekstinsyötön tehostamiseksi. Käyttäjän lii-  
kuttaessa sormiaan kosketuspinnalla, korostuu käyttäjän näkymään virtuaali-  
sessa näppäimistössä se kirjain, johon käyttäjä sillä hetkellä osuu. Lisäksi ohjai-  
met antavat käyttäjälle haptista palautetta. Ngn [2017] mukaan ratkaisu on  
nopeampi kuin tekstinsyöttö ohjaimilla osoittaen. Hänen mukaansa Viven oma  
ratkaisu, joka hyödyntää halkaistua näppäimistöä sekä kosketuspintoja, vaatii  
harjoittelua ja tuntuu epätarkalta.



Kuva 7. Ngn [2017] vaihtoehtoinen ratkaisu tekstinsyöttöön

---

Alun perin Googlen Daydream -projektin parissa kehitetyssä ”rumpunäppäimistössä” (drum keyboard) käyttäjä syöttää tekstiä mallintaen rumpusetillä soittamista. Ratkaisu on esitetty kuvassa 8. Virtuaalinen näppäimistö heijasteaan käyttäjän eteen viistosti, poiketen normaalisti heijastettavasta näppäimistöstä. Syöttääkseen tekstiä käyttäjä käyttää ohjaimia rumputikkuna, joilla käyttäjä ikään kuin lyö haluamiaan kirjaimia. Kuten laserosoitinratkaisussa, käyttäjä näkee rumputikkunsa ja sen kärkipisteen näkymässään. Näiden avulla käyttäjä pystyy kohdistamaan kärkipisteen haluamansa kirjaimen päälle. Googlen mukaan, verrattuna ohjaimien käyttämiseen pelkästään osoittimina, on rumpusetitratkaisu tehokkaampi tekstinsyöttöön. [Lopez 2016]. Ratkaisusta on poikunut avoimen lähdekoodin projekti nimeltä Cutie Keys, joka mallintaa Googlen ratkaisua.



Kuva

8. Rumpunäppäimistö [Lopez 2016]

---



## 4. Sekoitettu todellisuus

*Sekoitettu todellisuus* (engl. mixed reality) on yhdistelmä virtuaalitodellisuutta ja lisättyä todellisuutta (engl. augmented reality) [Brigham 2017]. Sekoitettu todellisuus eroaa virtuaalitodellisuudesta siten, että käyttäjä on yhtä aikaa tekemisissä virtuaalitodellisuuden sekä oikean todellisuuden kanssa. Tämä tarkoittaa sitä, että virtuaalisesti tuotetut esineet ja asiat voivat olla tekemisissä oikean maailman kanssa. [Claydon 2015]. Tällainen voisi olla virtuaalisesti luotu kirja, joka on asetettu oikeassa maailmassa sijaitsevan pöydän päälle.

### 4.1. Microsoft HoloLens

Microsoft HoloLens koostuu päähän puettavista HoloLens-laseista sekä mukana tulevasta kaukosäätimestä (clicker) [Microsoft 2017]. Samoin kuin virtuaalitodellisuuslaitteet, perustuu myös HoloLens-sensorien keräämään dataan. Lisäksi HoloLens käyttää kameroita käyttäjän päänliikkeiden seuraamisessa sekä ympäristön määrittelyssä. Lisäksi kamerat auttavat kädenliikkeiden seuraamisessa. [Colaner 2016]. HoloLens on esillä kuvassa 9.



Kuva 9. Microsoft HoloLens [Microsoft 2017]

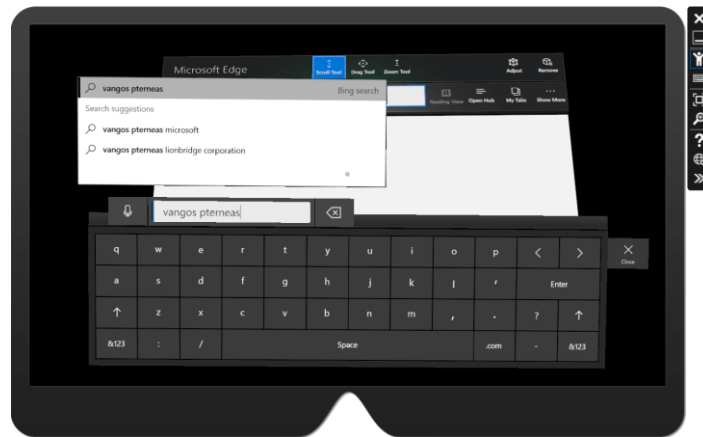
---

Toisin kuin virtuaalitodellisuuslaitteilla, HoloLensiä ei ohjata erillisillä ohjaimilla, vaan ohjaus perustuu käyttäjän eleisiin, katseeseen ja äänikomentoihin [Microsoft 2017]. Tämä tekee luontevasta tekstinsyötöstä vielä hankalampaa, ilman erillisiä lisälaitteita.

Koska Microsoft HoloLens ei käytä ohjaimia ympäristön hallintaan, vaan luottaa käyttäjän eleisiin ja katseeseen, on myös tekstinsyöttöön jouduttu kehittämään vaihtoehtoisia tapoja. Kuten esitellyissä virtuaalitodellisuuslaitteissa, myös HoloLens käyttää perinteistä virtuaalista näppäimistöä. Kuten muutkin esitellyt laitteet, HoloLens käyttää perinteistä QWERTY-mallia virtuaalisen näppäimistön pohjana. Näppäimistön yläreunasta löytyy kohta, jossa näkyy käyttäjän tämänhetkinen syöte.

Näppäimistöstä löytyy myös valmius syöttää erikoismerkkejä. Lisäksi näppäimistö sisältää ennakoivan tekstinsyötön. HoloLens sisältää valmiuden kah-

denlaiseen tekstinsyöttöön. Ensimmäinen vaihtoehto on käyttää katsetta sekä eleitä tekstinsyöttöön. Toinen vaihtoehto on syöttää tekstiä ääniohjauksella. [Microsoft 2017]. Ratkaisu on esitetty kuvassa 10.



Kuva 10. HoloLensin vakio tekstinsyöttö [Pterneas 2016]

Käyttäjän valitessa sekoitetun todellisuuden elementtejä, tapahtuu se katseella sekä valintaeleellä. Sama pätee myös tekstinsyöttöön. Kun ohjaimilla hallittavissa virtuaalitodellisuuksissa osoittimina toimivat ohjaimet, on HoloLensin osoitin käyttäjän katse. Käyttäjän näkymässä näkyy se kohta, johon katse sillä hetkellä osuu. Täten käyttäjä voi osoittaa haluamaansa kohdetta. Esillä olevasta virtuaalisesta näppäimistöstä käyttäjä osoittaa katseellaan kirjainta, jonka hän haluaa valita ja aktivoi sen eleellä. Eleenä toimii etusormen ja peukalon yhdistäminen ikään kuin napauttamalla. [Microsoft 2017]. Vaihtoehtona eleelle käyttäjä voi käyttää mukana tulevaa kaukosäädintä osoitetun kirjaimen valintaan [Rubino 2016].

Toinen vaihtoehto tekstinsyötölle on ääniohjaus. Käyttäjä valitsee virtuaalisesta näppäimistöstä sanelutilan (dictation mode) käynnistääkseen ääniohjauksen. Toteutus on varsin yksinkertainen. Käyttäjän puhuessa ääneen, siirtyy kaikki käyttäjän sanoma suoraan syötteeksi. Sanelutila tunnistaa myös yleisimpiä välimerkkejä. Esimerkiksi käyttäjän laushtaessa "Hey **comma** what are you up to **question mark**." muuttuu syöte muotoon "Hey, what are you up to?". Kun käyttäjä on valmis syötteensä kanssa, valitsee hän näppäimistön yläreunasta löytyvän done-näppäimen. [Microsoft 2017].

Molemmista ratkaisuissa on myös huonoja puolia. Moberg ja Petterson [2017] testasivat katseella ja eleellä kirjoittamista rakentamalla HoloLensin ratkaisua muistuttavan testitilanteen. Testauksessa verrattiin katseella ja eleellä kirjoittamista normaalilla kosketusnäytöllä kirjoittamiseen. Testauksessa kävi ilmi, että katseella ja eleellä kirjoittaminen on hitaampaa verrattuna koske-

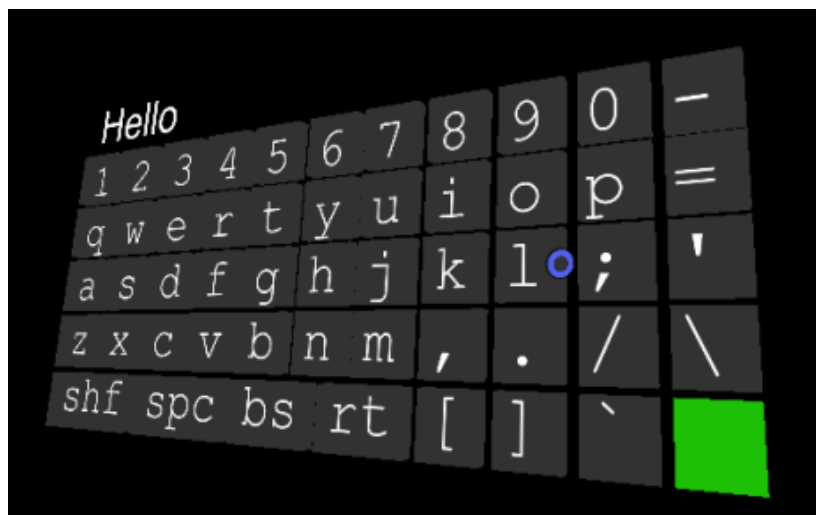
tusnäytöllä kirjoittamiseen. Osallistujien mielestä katseella ja eleellä kirjoittaminen on myös raskasta. HoloLensiä käyttäessä käyttäjän on myös mahdollista käyttää mukana tulevaa kaukosäädintä eleen sijasta, mutta tämä on vain pieni helpotus.

Tekstinsyöttö ääniohjauksella toimii hyvin, mutta sillä on myös luonnollisesti rajoitteita. Vaikeuksiin kuuluu esimerkiksi pitkissä saneluissa tavalliset puhuessa kertyvät ongelmat, kuten äänihuulten rasittuminen. Äänellä syöttämisen toinen vaikeus on ääntäminen. Järjestelmä ei välttämättä ymmärrä käyttäjän syötettä paksun aksentin tai huonon ääntämisen takia. [Finch 2017]. Äänellä syöttäessä kaikki syöte tulee sanella ääneen, joka ei luonnollisesti sovi kaikkiin tilanteisiin. Jos käyttäjä on esimerkiksi yleisissä tiloissa, ei hän välttämättä halua kaikkien kuulevan syöttämänsä sanomaa.

#### 4.2. HoloLensin vaihtoehtoiset tekstinsyöttötavat

Microsoft HoloLensille on tarjolla niukasti ilman lisälaitteita toimivia vaihtoehtoisia tekstinsyöttötapoja. Tarjolla olevat ratkaisut ovat lähinnä paranneltuja versioita virtuaalisista näppäimistöistä, joihin on lisätty ominaisuuksia helpottamaan tekstinsyöttöä ja käyttäjäkokemusta.

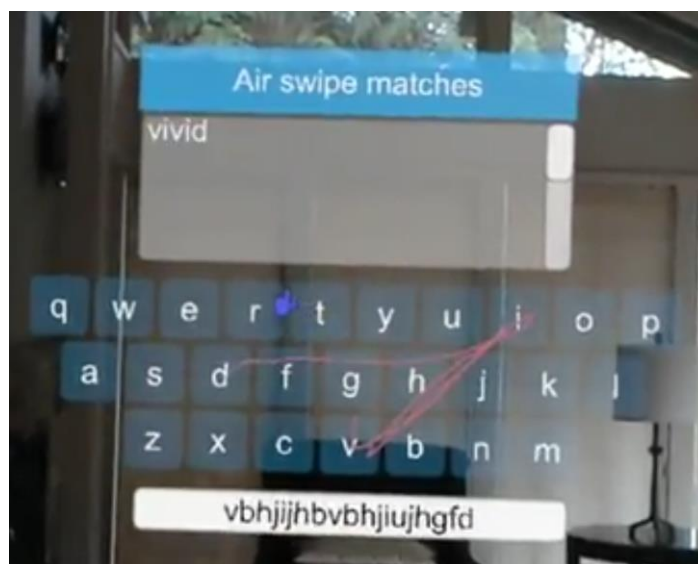
Microsoft-kaupassa tarjolla oleva Matrix Inceptionin kehittämä HoloLens D3D Keyboard on virtuaalinen näppäimistö, joka muistuttaa perinteistä virtuaalista näppäimistöä. Ratkaisu on esitetty kuvassa 11. Näppäimistö ei tarjoa tekstinsyötön kannalta ratkaisevia etuja verrattaen vakiotekstinsyöttöratkaisuun. Näppäimistö on suunniteltu pääasiassa HoloLensin parissa työskentelevien kehittäjien käyttöön eikä niinkään kuluttajille. [Matrix Inception 2017].



Kuva 11. HoloLens D3D [Matrix Inception 2017]

Marshwigggle Labsin kehittämässä virtuaalisessa näppäimistössä käytetään mobiilimaailmasta tuttua Swype-teknologian kaltaista ratkaisua. Swype on yksi teknologioista, joilla pyritään nopeuttamaan tekstinsyöttöä. Swypessä käyttäjä raahaa sormeaan kosketusnäytöllä jokaisen sanan sisältämän kirjaimen päältä. Sana päätetään nostamalla sormi pois näppäimistöltä. Ohjelmisto ennustaa sanaa jota käyttäjä on tavoitellut. [Swype 2017]. Marswigggle Labsin ratkaisu virtuaalisella näppäimistöllä toimii samoin. Sormen sijasta käyttäjä liikkuu katseellaan kirjaimien päältä ja ohjelmisto ennustaa niistä muodostuvaa sanaa. [Marswigggle Labs 2017]. Ratkaisu on esillä kuvassa 12. Näppäimistöstä ei ole paljoa tietoa tarjolla eikä sitä tarjota vielä avoimesti.

---



Kuva 12. Marswigggle Labs Airdoodle [Marswigggle Labs 2017]

---

Usein HoloLensillä suosituin ratkaisu tekstinsyöttöön on fyysinen näppäimistö. Koska sekoitetussa todellisuudessa käyttäjän koko visio ei ole peitetty, vaan käyttäjä voi helposti olla yhteydessä normaalin todellisuuden kanssa, on fyysisen näppäimistön käyttö luontevaa. Fyysistä näppäimistöä on mahdollista käyttää HoloLensin Bluetooth-teknologian kanssa [Rubino 2016].

## 5. Tulevaisuus

Laajennettujen todellisuuksien tekstinsyötön tulevaisuutta on hankala ennustaa. Sekä virtuaalitodellisuudessa että sekoitetussa todellisuudessa yksi seuraavista kehitysaskeleista on yksittäisten sormien seuraaminen. Tämä kehityssaskel helpottaisi huomattavasti myös tekstinsyöttöä ja loisi tilaa uudennlaisille ratkaisuille.

Microsoft on patentoinut ratkaisun, jossa käyttäjän eteen voidaan heijastaa näppäimistö, jolla käyttäjä voi kirjoittaa samoin kuin hän kirjoittaisi normaalilla näppäimistöllä. Microsoftin ratkaisussa sekoitettuun todellisuuteen heijastetaan virtuaalisesti luotu alusta, jolle on mahdollista heijastaa myös näppäimistö. Entistä tarkemman käsien seurannan sekä syvyyden hahmotuksen avulla käyttäjän on nyt mahdollista käyttää alustan päälle luotua näppäimistöä syöttäkseen tekstiä järjestelmään. Lopullisessa ratkaisussa käyttäjä näkisi virtuaalisesti luodun pöydän, jolle on heijastettu virtuaalisesti luotu näppäimistö, ja kirjoittaisi syötteensä tällä näppäimistöllä. Käyttäjä näkee näppäimistön ainoastaan laitteensa läpi ja näin ollen kirjoittaminen tapahtuu oikeasti joko ilmassa tai esimerkiksi oikeaa pöytää vasten, jolle alusta ja näppäimistö on heijastettu. [Enderle 2017].

Tarkkaa sormien jäljittämistä hakee myös Valve tulevalla ratkaisullaan. Valven kehityksen alla olevissa nyrkki-ohjaimessa (knuckles) pystytään entistä tarkempaan käsien ja sormien seurantaan. Ohjaimilla käyttäjä voi esimerkiksi tarttua virtuaalitodellisuuden elementteihin ja hallita niitä haluamansa mukaan. [Singletary 2017]. Ohjaimet avaavat luultavasti myös uusia mahdollisuuksia tekstinsyötön kannalta. Jos järjestelmä siten pystyy seuraamaan tarkasti kaikkia sormia, pitäisi myös tekstinsyöttö virtuaalisella näppäimistöllä onnistua luonnollisemmin.

Yksi vaihtoehto tekstinsyötölle laajennetuissa todellisuudessa on entistä pienemmät lisälaitteet. Tap-niminen yhtiö on kehittänyt sormien ympärille puettavan rannekkeen jolla käyttäjä voi kirjoittaa sormilla tehtävien eleiden avulla. Rannekkeen avulla käyttäjä syöttää tekstiä mitä tahansa pintaa vasten käyttäen sormilla tehtäviä eleitä. Rannekkeen voi liittää Bluetoothilla yleisimpiin käyttöjärjestelmiin ja siitä on povattu erityisesti lisälaitetta Microsoftin HoloLensille. [Boxall 2016]. Tuote on vasta ennakkotilattavissa.

Kaukaisemmassa tulevaisuudessa on ajatuksella ohjattava virtuaalitodellisuus. Jos käyttäjä voisi ohjata virtuaalitodellisuutta ajatuksillaan, luonnollisesti myös tekstinsyöttö olisi äärimmäisen helppoa. Startup-yritys Neurable kehittää järjestelmää, jossa käyttäjän on mahdollista hallita ympäristöään virtuaalitodellisuudessa ajatuksillaan. Neurable on onnistunut käyttämään ratkaisuaan yksinkertaisissa tehtävissä. [Metz 2017].

## 6. Yhteenveto

Tekstinsyöttö laajennetuissa todellisuuksissa on hankala kysymys johon ei ole vielä löytynyt kelpoa vastausta. Nyt käytettävistä ratkaisuista toimivin nojaa ohjainten sekä virtuaalisen näppäimistön käyttöön. Tehokkain tapa tekstinsyöttöön on joko käyttää ohjaimia laserosoittimina ja osoittaa ja vahvistaa

näkyvissä olevalta virtuaaliselta näppäimistöltä haluamansa kirjaimen tai käyttää ohjaimesta mahdollisesti löytyviä fyysisiä näppäimiä tai kosketuspintaa tekstin syöttämiseen.

Tutkielma antaa kuvan siitä, miten esitellyissä laitteissa tekstinsyöttö tapahtuu ja mitä vaihtoehtoja niille on. Joukon ulkopuolelle jää vielä osa laitteista, joita ei käsitelty tässä tutkielmassa. Paremman kuvan saavuttamiseksi tulisi käydä läpi kaikki tarjolla olevat laajennetun todellisuuden laitteet ja ratkaisut tekstinsyöttöön.

Ennen tarkkaa sormien jäljittämistä on yksi parhaista vaihtoehtoista tekstinsyötön helpottamiseksi käyttää hyväksi swype-teknologiaa. Siihen pohjautuvat virtuaaliset näppäimistöt helpottavat tekstinsyöttöä ja tulisi teknologiaa ottaa käyttöön myös nykyisissä ratkaisuissa. Kun sormien seuranta tarkentuu, joka mahdollistaa uudenlaiset ratkaisut tekstinsyöttöön, tulisi ottaa huomioon niitä tekijöitä, jotka vaikuttavat käyttäjäkokemukseen. Esimerkiksi kuinka ilmassa olevaan virtuaaliseen näppäimistöön kirjoittaminen eroaa normaalilla näppäimistöllä kirjoittamisesta.

Jatkotutkimuksissa tulisi testata parhaiksi todettuja vaihtoehtoja, jotta voitaisiin selvittää, mikä niistä on optimaalinen tekstinsyöttöön. Jatkotutkimuksia tulisi tehdä myös tarkemman seurannan, swype-teknologian ja sen parissa, kuinka nämä tekijät vaikuttavat käyttäjäkokemukseen. Tekstinsyöttö on oleellinen osa käyttäjäkokemusta laajennetuissa todellisuuksissa ja sen tutkiminen edesauttaa muita aloja, jotka käyttävät, tulevat käyttämään tai hyötyvät laajennetuista todellisuuksista.

## Viiteluettelo

- Nada Bates-Brkljac. 2012. *Virtual Reality*. Nova Science Publishers, Inc.
- Andy Boxall. *Get Ready to Change the Way You Type With This Amazing Wearable Keyboard*. <https://www.digitaltrends.com/mobile/tap-strap-wearable-keyboard-news/>. Checked 15.11.2017.
- Tara J. Brigham. 2017. Reality Check: Basics of Augmented, Virtual, and Mixed Reality. *Medical Reference Services Quarterly*. 36,2, 171-178.
- Mark Claydon. 2015. *Alternative Realities: From Augmented Reality to Mobile Mixed Reality*. M.Sc Thesis. School of Information Sciences. University of Tampere.
- Rob Enderle. The Virtual Keyboard and the Beginning of the End for Physical User PC Controls. <https://www.computerworld.com/article/3228986/microsoft-windows/the-virtual-keyboard-and-the-beginning-of-the-end-for-physical-user-pc-controls.html>. Checked 10.11.2017.

- Carol Finch. *The Disadvantages of Voice Recognition Software*. **Error! Hyperlink reference not valid.** [articles/the-disadvantages-of-voice-recognition-software](#). Checked 1.11.2017.
- Yongjae Lee and Gerard J. Kim. 2017. Vitty: Virtual touch typing interface with added finger buttons. In: *Proc. of VAMR 2017*, 111-118.
- Kif Leswing. *Here Are the Top 5 Most Popular VR Headsets — and What They Cost*. <http://www.businessinsider.com/vr-headsets-comparison-popularity-price-features-details-2017-6?r=US&IR=T&IR=T>. Checked 5.10.2017.
- Napier Lopez. *Google's VR Drum Keyboard Looks Like a Highly Inefficient But Fun Way to Type*. <https://thenextweb.com/google/2016/05/19/typing-drumsticks-vr-looks-terribly-inefficient-extremely-fun/>. Checked 9.11.2017.
- Marswiggle Labs. *Work*. <http://www.marshwigglelabs.com/work.html>. Checked 4.11.2017.
- Matrix Inception. HoloLens D3D Keyboard. <http://www.matrixinception.com/wp/products/hololens-d3d-keyboard/>. Checked 1.11.2017.
- Microsoft. *HoloLens*. <https://www.microsoft.com/en-us/hololens>. Checked 29.11.2017.
- Microsoft. 2017. *Why HoloLens*. <https://www.microsoft.com/en-us/hololens/why-hololens>. Checked 15.10.2017.
- Microsoft. *Hololens use gestures*. <https://support.microsoft.com/en-us/help/12644/hololens-use-gestures>. Checked 21.10.2017.
- Microsoft. *Hololens use your voice with hololens* <https://support.microsoft.com/en-us/help/12647/hololens-use-your-voice-with-hololens>. Checked 22.10.2017.
- Kyungha Min. 2011. Text Input Tool for Immersive VR Based on 3x3 Screen Cells. In: *Proc. of the 5th International Conference on Convergence and Hybrid Information Technology Lecture Notes in Computer Science vol 6935*, 778-786.
- William Moberg and Joachim Pettersson. 2017. *Interacting with Hand Gestures in Augmented Reality: a Typing Study*. Degree Project for Master of Science. Department of Creative Technologies. Blekinge Tekniska Högskola.
- Aaron Ng. *VR Test Input: Split Keyboard*. <https://medium.com/aaronn/vr-text-input-split-keyboard-e5bf3fd87a4c>. Checked 6.11.2017
- PCGamer. *SteamVR – Everything You Need to Know*. <http://www.pcgamer.com/steamvr-everything-you-need-to-know/>. Checked 10.10.2017.
- PCWorld. *HTC Vive*. <https://www.pcworld.co.uk/gbuk/tv-dvd-audio/gaming/virtual-reality/htc-vive-10144056-pdt.html>. Checked 29.11.2017.
- PlayStation. 2017. *Playstation Move Motion Controller*. <https://www.playstation.com/en-gb/explore/accessories/playstation-move-motion-controller/>. Checked 17.10.2017.

PlayStation. *Entering character*. <http://manuals.playstation.net/document/en/ps4/basic/osk.html>. Checked 20.10.2017.

PlayStation. *PlayStation VR*. <https://www.playstation.com/fi-fi/explore/playstation-vr/>. Checked 29.11.2017.

Lily Prasuehsut. *HTC Vive Review*.  
<https://www.wareable.com/vr/htc-vive-review>. Checked 11.10.2017

Vangos Pterneas. *Getting Started With HoloLens and Unity3D*.  
<https://pterneas.com/2016/04/04/getting-started-hololens-unity3d/>. Checked 29.11.2017

Daniel Rubino. *My first 24 hours with Microsoft HoloLens and awesome things I learned*. <https://www.windowscentral.com/my-first-24-hours-microsoft-hololens>. Checked 22.10.2017.

Charles Singletary. *See How Valve's Knuckles Prototype Works In This Demo Video*.  
<https://uploadvr.com/valve-knuckles-demo-video/>. Checked 17.11.2017.

Statista. *Forecast Augmented (AR) and Virtual Reality (VR) Market Size Worldwide From 2016 to 2021 (In Billion U.S Dollars)*. <https://www.statista.com/statistics/591181/global-augmented-virtual-reality-market-size/>. Checked 5.10.2017.

Swype. *Android Features* <http://www.swype.com/product-features/android/features.html>. Checked 3.11.2017.

Vive. *Vive Virtual Reality System*. <https://www.vive.com/us/product/vive-virtual-reality-system/>. Checked 10.10.2017.

Bonnie Yu. *Search in VR*. [http://bonnieyu.com/vr\\_search\\_explore.html](http://bonnieyu.com/vr_search_explore.html). Checked 20.10.2017.



# Pörssikauppaa käyvät algoritmit

Otto-Ville Savolainen

## Tiivistelmä.

Tämän tutkimuksen tarkoituksena on tutustua algoritmisen kaupankäynnin kehityskaareen aina alkuajoista tulevaisuuteen saakka sekä selvittää nykyisin yleisimmin käytettyjä algoritmisen kaupankäynnin strategioita ja strategioihin liittyviä algoritmeja. Tutkimuksessa käsiteltäviä strategioita on kolme ja jokaisesta kategoriasta esitetään yksi kategorialle ominainen algoritmi. Kyseisen algoritmin toimintaa pyritään havainnollistamaan käytännönläheisten esimerkkien avulla.

Tämä tutkimus on kirjallisuuskatsaus. Omia näkemyksiä ja pohdintoja pyritään tuomaan mukaan monipuolistamaan tekstin sisältöä ja antamaan lukijaläheisemmän lähestymistavan tutkimukselle.

Algoritmisen kaupankäynti on etenkin erilaisten arvopapereita välittävien yritysten ja pankkien yleisessä käytössä oleva tapa tuoton maksimoimiseksi, mutta yksityisiä sijoittajia algoritmit eivät ole vielä tavoittaneet. Kaupankäyntialgoritmeja pyörittävien tietokoneiden laskentatehon on aiemmin nähty vaikuttavan merkittävästi algoritmin tehokkuuteen, mutta tutkimuksen perusteella sen roolin nähdään vähenevän tulevaisuudessa. Laskentatehoa merkittävämpään rooliin nousee tekoäly ja neuroverkot, joiden avulla algoritmit saadaan toimimaan entistä tehokkaammin ja älykkäämmin, vaikka varsinaista laskentatehoa ei lisätäisikään.

**Avainsanat ja -sanonnat:** algoritmisen kaupankäynti, pörssikauppa, tietokoneavusteinen kaupankäynti, osakekauppa, osakekaupankäynti

## 1. Johdanto

Tämä tutkimus alkaa algoritmisen kaupankäynnin yleisellä esittelyllä luvussa 2. Kyseisessä luvussa määritellään algoritmisen kaupankäynti, tarkastellaan algoritmisen kaupankäynnin kehitystä aina 1970-luvulta tähän päivään asti sekä pyritään arvioimaan tulevaisuuden näkymiä. Luvussa 3 esitellään vaikutusvetoisten algoritmien pääpiirteet ja paneudutaan yhteen kaupankäyntistrategiaan, volyymipainotteiseen keskihintaan, hieman tarkemmin. Luku 4 koostuu opportunististen algoritmien peruseräiteiden esittelystä. Samassa luvussa syvennyttään parikaupankäyntiin hieman tarkemmin. Luku 5 puolestaan käsittelee

hintavetoisten algoritmien pääpiirteitä. Hintavetoisista algoritmeista esitellään toteutus-  
vaje. Viimeinen luku koostuu koko tutkielmaa koskevasta yhteenvedosta, jossa kerrataan  
hieman tärkeimpiä kohtia aiemmin esitetyistä asioista.

Algoritmeja käsittelevissä luvuissa annetut esimerkit ovat yleisessä käytössä olevia kau-  
pankäyntistrategioita, joita pystytään hyödyntämään myös ilman algoritmeja. Nämä strate-  
giat ovat kuitenkin niin yleisiä ja helposti automatisoitavia, että niiden käyttäminen algorit-  
mien avulla on järkevää. Tämän vuoksi nämä kaupankäyntistrategiat esitetään tutkimuk-  
sessa algoritmisesta näkökulmasta katsottuna, vaikka kyseessä ei ole varsinaisesti osake-  
kaupankäyntiin suunniteltu algoritmi.

## 2. Algoritminen kaupankäynti

Suomen Pörssisäätiön [2017a] mukaan osakekaupankäynti perustuu kaupankävijän ja vä-  
littäjän väliseen vuorovaikutukseen, jossa kaupankävijä antaa välittäjälle toimeksiantoja,  
jotka välittäjä rekisteröi pörssin kaupankäyntijärjestelmään. Toimeksiantotyypppejä on mo-  
nenlaisia, mutta yksinkertaisimmillaan toimeksianto voi koostua esimerkiksi osakkeen os-  
tosta tai myynnistä. Tässä tapauksessa puhutaan niin sanotusta *osto- tai myyntimääräyksestä*  
(market order), jossa osaketta ostetaan tai myydään sillä hetkellä parhaalla mahdollisella  
hinnalla. Muita yleisiä toimeksiantotyypppejä ovat muun muassa *rajahinta* (limit order) ja  
*tappionpysäytys* (stop order). Rajahinnassa toimeksiannolle määritellään hinta, jota kalliim-  
malla osaketta ei haluta ostaa, tai vastaavasti jota halvemmalla osaketta ei haluta myydä.  
Tappionpysäytyksessä ideana on suojata kaupankävijää tappioilta määrittämällä osake-  
keelle hinta, jossa se myydään, mikäli sen arvo laskee määritellylle tasolle. [Labadie and  
Lehalle 2010.]

*Algoritmiselle kaupankäynnille* (algorithmic trading) on useita määrittelyjä, mutta Buschin  
[2016] mukaan algoritmisesta kaupankäynnistä voidaan puhua silloin, kun tietokoneelle  
syötetty algoritmi määrittelee täysin ilman ihmistä tai enintään rajoitetulla ihmisen väliin-  
tulolla toimeksiannolle tarvittavat parametrit. Tämän lisäksi määrittelyyn kuuluu algorit-  
min päätöksentekokyky, eli kuinka toimeksiannon jättämisen jälkeen toimitaan. Parametrit  
ovat arvoja, joiden perusteella algoritmi toimii ja tekee laskelmansa. Esimerkiksi toimeksi-  
annon tyyppi, kohteena toimiva osakeyhtiö, tarkkailtava aikaväli ja osto-/myyntihinta voi-  
sivat toimia parametreina. Suuri osa algoritmien käyttämisestä parametreista on vielä ihmisen  
määrittelemiä, mutta kehittyneimmät algoritmit osaavat hakea itse itselleen tietyllä hetkellä  
parhaaseen tulokseen johtavat parametrien arvot. [Leshik and Cralle 2011.]

### 2.1. Algoritmisen kaupankäynnin kehitys

Vielä 1970-luvulle asti toimeksiannot tuli tehdä manuaalisesti välittäjän kautta, kunnes  
New Yorkin pörssi (The New York Stock Exchange, NYSE) kehitti vuonna 1976 täysin uu-  
den kaupankäyntijärjestelmä DOT:n (designated order turnaround), jota seurasi SuperDOT

vuonna 1984. SuperDOT:n avulla toimeksiannot pystyttiin lähettämään sähköisinä suoraan kauppapaikalle, jossa asiantuntijat toimeenpanivat niitä. Samoihin aikoihin maailman ensimmäinen täysin sähköinen arvopaperipörssi NASDAQ (National Association of Securities Dealers Automated Quotations) alkoi vakiinnuttaa asemaansa Yhdysvaltojen pörssi-markkinoilla. [Markham and Harty 2008.]

Sähköisen viestinnän verkot (electronic communications networks, ECN) kehittyivät 1990-luvun puolivälissä helpottamaan ja monipuolistamaan arvopaperikaupankäyntiä. ECN:t ovat pääsääntöisesti yksityisiä verkkoja, jotka noudattavat peruspiirteiltään samanlaista strategiaa: rekisteröidään asiakkaan toimeksianto sisäiseen rekisteriin ja toimeenpannaan se automaattisesti silloin, kun toimeksiannolle löytyy vastine sille annettujen piirteiden perusteella (eli ostajalle myyjä ja päinvastoin). Toimeksiannot asetetaan tärkeysjärjestykseen suoraan ajan ja hinnan mukaan. Osto- ja myyntimääräykset priorisoidaan korkeimmaksi, kun taas rajahinta toimeksiantojen prioriteetti määräytyy asiakkaan määrittelemän hinnan mukaan. Asiakas, joka on valmis ostamaan korkeimmalla hinnalla tai vastaavasti myymään alhaisimmalla hinnalla, saa korkeimman prioriteetin. Kahden samanlaisen rajahinta toimeksiannon tapauksessa aiemmin vastaanotettu toimeksianto saa korkeamman prioriteetin. Mikäli korkeimmalla prioriteetilla olevalle toimeksiannolle ei löydy vastinetta, ECN:n avulla se voidaan lähettää julkiseen pörssiin NASDAQ:iin. [Liebenberg 2002.]

Kim [2007] toteaa, että vuonna 2001 Yhdysvaltojen arvopaperimarkkinoita valvova elin SEC (United States Securities and Exchange Commission) määräsi desimalisaation eli osakkeiden hinnat tuli jatkossa ilmoittaa desimaaleina 0,01\$ tarkkuudella murtolukujen sijaan. Aiemmin pienin osakkeen hintayksikkö oli  $1/16\$$ , eli 0,0625\$. Desimalisaation taustalla oli ajatus alentaa kaupankäyntikuluja, tehdä osakkeiden hinnat ymmärrettävämmiksi ja näin ollen helpottaa myös piensijoittajien tuloa osakemarkkinoille. Desimalisaation seurauksena suurten sijoittajien suosima marginaalikaupankäynti kuitenkin heikkeni ja heidän osallisuutensa markkinoilla väheni. Jäljelle jääneet sijoittajat omaksuivat sähköiset kaupankäyntijärjestelmät pienten kaupankäyntikulujen vuoksi nopeasti, minkä seurauksena osakkeita välittävien osapuolien kilpailuetu heikkeni.

Aiemmin esiteltyjen tapahtumien lisäksi myös yleisen informaatioteknologisen kehityksen niin internetin kuin tietokoneidenkin osalta voidaan nähdä vaikuttaneen positiivisesti algoritmisen kaupankäynnin kasvuun [Leshik and Cralle 2011]. Lisäksi on mahdollista, että 2000-luvun alun desimalisaatiosta johtuva 0,01\$ porrastus osakkeiden hinnoissa johti algoritmisen kaupankäynnin kehittymiseen myös siitä syystä, että algoritmit voidaan määritellä hyödyntämään pienetkin hinnanmuutokset mahdollisen tuoton takaamiseksi.

## **2.2. Algoritmisen kaupankäynti nykyään**

Nykyään algoritmit ovat käytössä lähes jokaisessa arvopapereita välittävässä instituutiossa, ja niiden taustalla toimii niin matematiikan, tilastotieteen, fysiikan kuin IT-alankin asiantuntijoita sekä algoritmeja suorittavat supertietokoneet. Algoritmista kaupankäynnistä on

tullut pankkien sekä muiden arvopapereita välittävien yritysten yksi pääasiallisista kilpailukeinoista, ja Yhdysvaltojen osakemarkkinoilla algoritmisesta kaupankäynnistä on tullut valtavirtaa: Leshikin ja Crallen [2011] mainitseman Citigroupin raportin mukaan yli puolet kaikista Yhdysvaltojen osakekaupoista on algoritmien käsittelemiä. Vaikka monet instituutiot käyttävät pääpiirteittäin samanlaisia algoritmeja, jokainen käyttäjä muokkaa algoritmin tekemään sille tarkoitetun tehtävän käyttäjälle sopivalla ja tilanteeseen nähden parhaalla mahdollisella tavalla. Tämän vuoksi instituutiot suojelevat algoritmien lähdekoodeja tarkasti. [Leshik and Cralle 2011.] Lattman [2010] kertoo yhdysvaltalaisen liikepankin Goldman Sachs'n entisen ohjelmoijan Sergey Aleynikovin varastaneen noin 300 miljoonaa dollaria vuodessa tuottaneen algoritmin, kun kilpailija tarjosi Aleynikoville kolminkertaisen (1,2 miljoonaa dollarin) vuosipalkan verrattuna Goldman Sachsiin. Tämä tapahtumaketju kuvastaa erittäin hyvin sitä, millainen rooli ja volyyymi algoritmisella kaupankäynnillä on nykyään osana pankkien ja muiden arvopaperivälittäjien liiketoimintaa.

Arvopaperikaupankäynnissä tapahtunut nopea teknologinen kehitys johti Euroopan unionin (EU) luomaan rahoitusvälineiden markkinat -direktiivin (Markets in Financial Instruments Directive, MiFID) vuoden 2007 loppupuolella [Lenglet 2001]. Vuonna 2014 säädettiin uusi direktiivi MiFID II, jonka on määrä astua voimaan vuoden 2018 alusta alkaen. Tämä uudistettu direktiivi ottaa kantaa algoritmiseen kaupankäyntiin ja antaa määräykset sille sekä siihen läheisesti liittyviin termeihin. Direktiivin näkökanta algoritmiseen kaupankäyntiin asettuu keskiviivalle: direktiivi ei kiellä algoritmista kaupankäyntiä, mutta se vaatii sitä harjoittavilta toimijoilta tarkoin määrättyä valvontaa algoritmisen kaupankäynnin tuomien riskien välttämiseksi. Tarkoin määritelty valvonta koskee muun muassa algoritmista kaupankäyntiä harjoittavien yritysten sisäistä järjestelmää ja sen hallintaa, sekä yrityksen tietoja ja asiakirjoja, joista käy ilmi esimerkiksi heidän algoritmiset kaupankäyntistrategiat ja algoritmien käyttämät parametrit. [Busch 2016.]

Algoritmisen kaupankäynti on vielä pääosin pankkien ja arvopapereita välittävien yritysten kilpapakenttä, sillä algoritmeja hyödyntäviä yksityisiä sijoittajia on murto-osa kaikista markkinoilla olijoista. Tämä johtuu osittain algoritmeihin liittyvästä tiedon puutteesta sekä niiden heikosta saavutettavuudesta. Yksityisiä sijoittajia kuitenkin pyritään saamaan algoritmien käyttäjiksi tarjoamalla niihin liittyvää kirjallisuutta ja simulaattoreita. Näiden avulla yksityiset sijoittajat voivat saada tietoa ja opastusta algoritmeista sekä kokeilla algoritmien toimintaa ilman pelkoa taloudellisista menetyksistä. [Leshik and Cralle 2011.]

### **2.3. Algoritmisen kaupankäynnin tulevaisuus**

Tietokoneiden laskentateho kehittyi entistä automatisoivammaksi, halvemmaksi ja nopeammaksi vuosi vuodelta. Vuodesta 1929 vuoteen 2009 asti Yhdysvaltojen pörssimarkkinoiden kapitalisointi on tuplaantunut jokaisella vuosikymmenellä. Samalla aikavälillä kaupankäyntivolyyymi Dow Jones Industrial Averagesissa on tuplaantunut 7,5 vuoden välein, mutta

viimeisen vuosikymmenen aikana tuplaantuminen on tapahtunut jo vajaassa kolmessa vuodessa. Vaikka edellä mainitut kehityssuunnat ovat epäilemättä hyötyneet Mooren laista, niin muutokset ihmisten käyttäytymisessä vaikuttaa tietokoneiden laskentatehoa enemmän finanssialan kehitykseen. [Kirilenko and Lo 2013.]

Mooren laki on prosessorivalmistaja Intelin perustajan Gordon E. Mooren tekemä havainto vuonna 1965, jonka mukaan komponenttien määrä mikrosiruissa tuplaantuu joka vuosi. Kymmenen vuotta havainnoituaan prosessorien kehitystä Moore muutti ennustettaan tuplaantumisen tapahtuvaksi joka toinen vuosi. Tämäkin ennuste oli hieman liian pessimistinen, joten myöhemmin ajanjakso muutettiin vielä 18 kuukaudeksi. Jo 50 vuotta orjallisesti sääntöään noudattaneen Mooren lain epäillään tulevansa piakkoin tiensä päähän. [Mack 2011.] Voidaan siis päätellä, ettei algoritmisen kaupankäynnin tulevaisuudessa kilpailla enää niinkään algoritmeja suorittavien tietokoneiden prosessorien tehokkuudesta ja algoritmien nopeudesta, vaan pikemminkin käyttäjien algoritmisesta ymmärryksestä ja kyvystä oppia hyödyntämään algoritmeja järkevästi.

Toisaalta Leshik ja Crane [2011] mainitsevat algoritmeihin kohdistuvan massiivisia investointeja, joiden avulla algoritmeista tehdään entistä älykkäämpiä ja itseohjautuvampia. Tähän teemaan liittyy vahvasti tekoälyn ja neuroverkkojen kehitys sekä kysymys siitä, kuinka niitä voitaisiin hyödyntää algoritmisessa kaupankäynnissä. Tekoälyn kehittyessä ennustettavuus, assosiaatioiden löytäminen ja kaupankäyntistrategioiden generoiminen korostuvat pörssikaupassa, kun suuria määriä erilaisia algoritmisia variaatioita kyetään testaamaan rinnakkain samalla yhdistellen optimaalisia ratkaisuja. Tämän lisäksi ihmisten rooli itse algoritmien kehityksessä mahdollisesti kutistuu, kun tekoäly luo, testaa ja hienosäätää uusia algoritmeja sekä niille annettavia parametreja. [Johnson 2010.]

Myös big data tulee näyttelemään aiempaa suurempaa roolia algoritmisessa kaupankäynnissä ja osakekaupassa yleisestikin. Tiedonloughinnalla pyritään etsimään toistuvia trendejä, joiden avulla voidaan tehdä sääntöjä ja yhteyksiä asioiden ja tapahtumien välille. Tiedonloughinta voidaan jakaa kolmeen eri tyyppiin, joita ovat klusterointianalyysi, aikasarjojen loughinta ja assosiaatiosääntöjen loughinta. [Johnson 2010.]

Johnson [2010] määrittelee klusterointianalyysin loughinnaksi, jonka avulla datasta etsitään yhtäläisyyksiä eri havaintojen väliltä. Kun yhtäläisyyksiä löytyy, ennusteet voidaan tehdä käsittelemään kaikkia vastaavilla piirteillä olevia havaintoja. Oletetaan klusterointianalyysin löytäneen samankaltaisuuksia esimerkiksi jäätelöä valmistavista yrityksistä. Jos yhden jäätelövalmistajan myynti kasvaa aina kesäisin, ennusteen perusteella näin voidaan olettaa tapahtuvan myös muiden jäätelövalmistajien myynnissä.

Aikasarjojen loughinta on nimensä mukaisesti loughintaa, jossa keskitytään tarkastelemaan osakkeen hinnan muutoksia eri ajanjaksoina ja pyritään löytämään ajallisia sääntöjä [Johnson 2010]. Jäätelöt toimivat edelleen hyvänä esimerkkinä, sillä aikasarjojen loughinnan

avulla voitaisiin huomata jäätelövalmistajien myynnin kasvavan merkittävästi aina kesäisin.

Viimeisenä tiedonlouhintatyyppinä on assosiaatiosääntöjen louhinta, jonka avulla pyritään löytämään assosiaatioita kahden erilaisen alkion väliltä. Esimerkkinä toiminee monelle tuttu ostoskori-esimerkki, jossa kaupat etsivät tietokannoista eri tuotteiden välisiä yhteyksiä [Johnson 2010]. Assosiaatiosääntöjen louhinnan avulla voidaan huomata esimerkiksi jäätelön A ja mansikkahillon B välillä yhteys, jolloin jäätelön A kohonneen myynnin ansiosta voidaan olettaa myös mansikkahillon B myynnin kohoavan.

Vaikka aiemmin esitetyt esimerkit eivät suoranaisesti liity osakekaupankäyntiin, samoja louhintastrategioita voidaan hyödyntää myös pörssikaupassa louhimalla osakkeisiin liittyvää dataa. Käsittelimieni suuntauksien perusteella voidaan kuitenkin todeta, ettei algoritmisen kaupankäynnin tulevaisuuden näkymät ole helposti ennustettavissa. Teknologinen kehitys on jo niin pitkällä, että jokaista käsittelemääni ilmiötä hyödynnetään jo osakekaupassa tavalla tai toisella. Yleinen näkemys lienee kuitenkin se, että nämä ilmiöt ovat vielä kehityksensä alkutaipaleella ja kehitys näissä ilmiöissä voidaan nähdä johtavan lähes poikkeuksetta kehitykseen myös algoritmisessa kaupankäynnissä.

### 3. Vaikutusvetoiset algoritmit

*Vaikutusvetoisten algoritmien* (impact-driven algorithms) ideana on pilkkoa suuret toimeksiannot useiksi pieniksi toimeksiannoiksi, jolloin toimeksiannon toteutuessa vaikutus osakkeen hintaan pysyy matalana ja kaupankäynnistä syntyvät kokonaisvaikutukset saadaan minimoitua. Nimensä mukaisesti vaikutusvetoisten algoritmien päätarkoituksena on siis pitää vaikutukset markkinoihin mahdollisimman pieninä. Vaikutusvetoisissa algoritmeissa kaksi algoritmia nousee selkeästi yli muiden: aikapainotettu keskihinta (time weighted average price, TWAP) ja volyymipainotteinen keskihinta (volume weighted average price, VWAP). Nämä strategiat edustavat vaikutusvetoisten algoritmien ensimmäistä sukupolvea ja ovat pääosin aikataulupohjaisia algoritmeja, jonka vuoksi niiden kehittymiseen ei juurikaan vaikuta markkinaolosuhteet, kuten hinta tai volyymi. Niiden tehtävänä on vain toteuttaa toimeksianto parametrina annettuna aikajaksona. [Johnson 2010.]

#### 3.1. Volyymipainotteinen keskihinta

Volyymipainotteinen keskihinta on yksi vanhimmista, mutta edelleen suosituimmista algoritmisessa kaupankäynnissä käytettävistä algoritmeista. Sen suosion nähdään perustuvan kolmeen tekijään. Ensinnäkin, volyymipainotteinen keskihinta on helppo laskea, mikä heijastuu myös helppoon kaupanjälkeiseen raportointiin. Toiseksi, volyymipainotteinen keskihinta edistää suurten toimeksiantojen pilkkomista osiin, jolloin se vähentää tarvetta korkealle maksukyvyille ja näin ollen pienentää myös vaikutusta markkinoihin. Berkowitz ja muut [1988] muotoilivat: "The volume-weighted average price on any day represents the

price a "naive" trader can expect to obtain." Tämä tarkoittaa, että volyymipainotteista keskihintaa voidaan käyttää kelvollisena vertailukohtana tietyssä aikavälillä osakkeen hinnalle, mikä on myös viimeinen syy sille, miksi volyymipainotteinen keskihinta on saavuttanut suosionsa yhtenä algoritmisen kaupankäynnin käytetyimpänä algoritmina. [Frei and Westray 2013.]

### 3.2. Volyymipainotteisen keskihinnan toimintaperiaate

Kuten algoritmin nimikin jo hieman enteilee, VWAP kuvastaa tietyn osakkeen kokonaismyyntiä jaettuna kokonaismyyntivolyymilla tietyssä ajajaksossa. Matemaattisesti tämä voidaan esittää muodossa

$$VWAP = \frac{\sum_n V_n P_n}{\sum_n V_n},$$

jossa  $n$  on osakekauppojen kokonaismäärä,  $V$  on yhden osakekaupan volyymi ja  $P$  on osakkeen hinta yhdessä osakekaupassa. Toisin kuin aikapainotteisessa keskihinnassa, volyymipainotteisessa keskihinnassa tietyssä ajajaksossa tapahtuva kaupankäynti vaatii oikean kokoisen toimeksiannon. Koska tämä toimeksiannon oikea koko riippuu osakkeen yhden päivän kokonaiskaupankäyntivolyymista, jota ei voi tietää etukäteen, on osakkeen kaupankäyntiä syytä tarkastella historiallisesta näkökulmasta. Jos yksi päivä koostuu  $j$ :stä ajajaksosta, päivittäinen VWAP voidaan ilmaista kaavalla

$$VWAP = \sum_j u_j \bar{p}_j,$$

jossa  $u$  on prosentuaalinen osuus päivän kaupoista ja  $\bar{p}$  on osakkeen keskihinta kullakin ajajaksolla. Tämän perusteella siis optimaalinen määrä toimeksiantoja  $x$  tietyssä ajajaksossa  $j$  on

$$x_j = u_j X,$$

kun  $X$  on alkuperäisen toimeksiannon koko. Tämä kaava toistuu suurimmassa osassa volyymipainotteisen keskihinnan algoritmien variaatioissa. [Johnson 2010.]

Havainnollistetaan tilannetta mukauttamalla hieman Lin ja Yen [2013] tarjoamaa esimerkkiä. Oletetaan, että yhtenä päivänä ostetaan 1000 kappaletta yhtä osaketta viiden tunnin aikana. Taulukosta 1 löytyy keksimiäni arvoja, joiden avulla ilmenee kunakin ajajaksossa ostettujen osakkeiden määrä ja hinta. Näiden avulla pystytään laskemaan sen päivän volyymipainotteinen keskihinta kyseiselle osakkeelle. Hyödyntämällä aiemmin mainittua kaavaa saadaan kyseisen päivän volyymipainotteiseksi keskihinnaksi

$$VWAP = \frac{300 \cdot 50 + 200 \cdot 51 + 100 \cdot 50,5 + 100 \cdot 50 + 300 \cdot 51}{300 + 200 + 100 + 100 + 300} = 50,55.$$

Ajanjakso (tunti)	Osakkeen hinta	Ostovolyyymi (% kokonais- volyymistä)	Ostettuja osakkeita yhteensä
1	50	300 (30%)	300
2	51	200 (20%)	500
3	50,5	100 (10%)	600
4	50	100 (10%)	700
5	51	300 (30%)	1000

Taulukko 1. Esimerkkitaulukko volyymipainotteisen keskihinnan muodostumisesta.

Seuraavan päivänä halutaan ostaa uudestaan samaa osaketta, mutta tällä kertaa 10 000 kappaletta viiden tunnin aikana. Tarkastellaan siis historiallisesta näkökulmasta (tässä tapauksessa edellisen päivän) osakkeen volyymipainotteista keskihintaa. Kunakin ajanjaksona ostetaan siis

$$x_1 = 0,3 \cdot 10000 = 3000 \text{ osaketta}$$

$$x_2 = 0,2 \cdot 10000 = 2000 \text{ osaketta}$$

$$x_3 = 0,1 \cdot 10000 = 1000 \text{ osaketta}$$

$$x_4 = 0,1 \cdot 10000 = 1000 \text{ osaketta}$$

$$x_5 = 0,3 \cdot 10000 = 3000 \text{ osaketta.}$$

Vastaavaa tapaa noudattaen algoritmi kykenee tarkastelemaan osakkeen historiaa pidemmältä aikaväliltä ja vaikeammilla luvuilla. Johnson [2010] toteaa, että VWAP-algoritmit saattavat jossain tapauksessa mennä annetun aikataulun edelle tai jäädä siitä jälkeen, mutta modernit algoritmit kykenevät itse tekemään tällaisissa tapauksissa päätökset, jotka johtavat parhaaseen lopputulokseen. Niiden suorituskyky perustuu sekä kohdeosakkeen seurantakykyyn että markkinavolyymien ennakoimiseen. Tästä syystä historialliset profiilit voivat olla merkittävässä roolissa algoritmin toimintaa. Lisäksi, koska VWAP on herkkä äkkinaisille muutoksille markkinoilla, osa algoritmeista on määritelty tarkkailemaan myös markkinoiden nykyistä tilannetta kyetäkseen mukautumaan muuttuvaan tilanteeseen dynaamisesti.



## 4. Opportunistiset algoritmit

*Opportunistiset algoritmit* (opportunistic algorithms) ovat kehittyneet eri kaupankäyntistrategioiden seurauksena. Niiden perusajatuksena on nimensäkin mukaisesti käyttää markkinoilla vallitsevaa tilannetta hyödykseen, vaikutti markkinatilanteeseen sitten osakkeen hinta, maksuvalmius tai mikä tahansa muu tekijä. Opportunistisia algoritmeja on erilaisia, joista esimerkiksi osa keskittyy juuri hintaan ja osa maksuvalmiuteen. Vaikka kullakin algoritmilla on oma strategiansa, hinta on silti vahva tekijä jokaisen opportunistisen algoritmin toiminnassa. Algoritmille syötetyn kohdeosakkeen avulla saadaan vertailukohta, jolloin algoritmi kykenee laskemaan, onko markkinaolosuhteet otolliset vai eivät. [Johnson 2010.]

### 4.1. Parikaupankäynti

*Parikaupankäynti* (pairs trading) on kaupankäyntistrategia, jota käytetään hyväksikäyttämään epätasapainossa olevaa markkinatilannetta. Goldman Sachsin mukaan epätasapainossa olevat markkinat päätyvät järkevään tasapainoon ennemmin tai myöhemmin, ja kaupankävijällä on kiinnostusta saada maksimaalinen hyöty epätasapainon aiheuttamasta poikkeavuudesta. [Elliot *et al.* 2005.] Song ja Zhang [2013] toteavat parikaupankäynnin neutraalin luonteen olevan sen etu. Tällä tarkoitetaan sitä, että parikaupankäynnin avulla pystytään tekemään tuottoa markkinaolosuhteista riippumatta.

#### 4.1.1. Parikaupankäynnin toimintaperiaate

Parikaupankäynnissä algoritmin ideana on käsitellä osakkeista saatavilla olevaa historiallista dataa ja etsiä sen avulla kaksi toisiaan vahvasti korreloivaa osaketta. Tällaisen osakeparin löytyessä niitä tarkkaillaan, kunnes niiden hinnat alkavat erota toisistaan (eli toisen hinta nousee ja toisen laskee), jolloin parikaupankäynti käynnistyy. Kun osakkeiden hinnat alkavat palata lähelle toisiaan, osakkeet realisoidaan. [Song and Zhang 2013.]

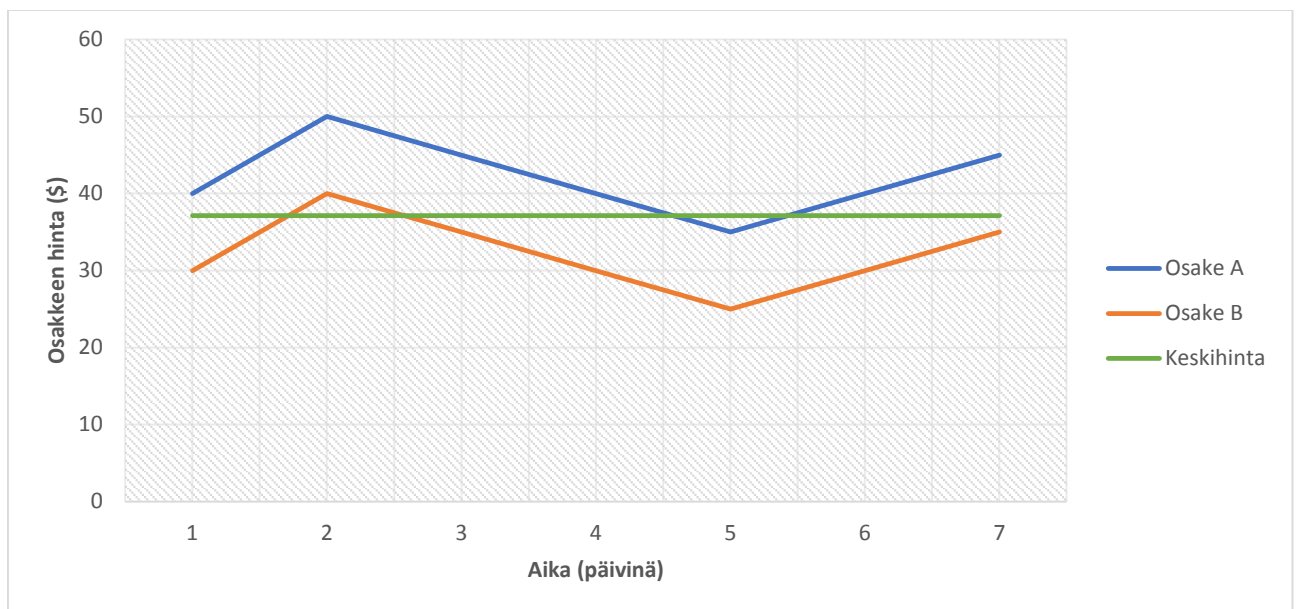
Toisiaan vahvasti korreloivat osakkeet voidaan ajatella löytyvän usein saman toimialan sisältä, sillä tällaisessa tapauksessa molempiin osakkeisiin vaikuttavat samat markkinaolosuhteet. Se, minkä takia osakkeiden hinnat alkavat erota toisistaan, voi johtua monista syistä. Yksi yleinen syy on esimerkiksi vain toiseen yritykseen kohdistuvat uutiset [Engelberg *et al.* 2008]. Ajatellaan tilannetta esimerkin kautta. Oletetaan kahden vaatteita myyvän yrityksen A ja B osakkeiden korreloivan vahvasti toisiaan. Jos yritys A jää kiinni lapsityövoiman käytöstä, asiakkaiden voidaan olettaa siirtyvän yrityksen B asiakkaiksi. Tämän tapahtuman seurauksena yrityksen A osakkeen hinnan voidaan olettaa laskevan ja yrityksen B puolestaan nousevan.

Zeng ja Lee [2014] tiivistivät parikaupankäynnin kolmeen helposti ymmärrettävään askeleeseen. Ensiksi valitaan osakepari ja lasketaan näiden kahden osakkeen hinnan suhteen keskiarvo ja keskihajonta. Kun suhde poikkeaa keskiarvosta kahdella keskihajonnalla, yli-

*myydään* nousevaa osaketta ja ostetaan laskevaa osaketta. Kun suhde palaa lähelle keskiarvoa, realisoidaan osakkeet ja kerätään voitot. Suomen pörssisäätiö [2017b] määrittelee ylimyynnin tarkoittavan tilannetta, jossa kaupankävijä lainaa arvopapereita ja myy ne, vaikkei varsinaisesti omistakaan niitä. Tämän avulla kaupankävijä pyrkii hyötymään kurssien laskusta, sillä hän olettaa saavansa ostettua ylimyymänsä osakkeet myöhemmin halvemmalla.

Algoritmin haasteena on määritellä sopivat arvot, jotka käynnistävät parikaupankäynnin (kuinka suuri ero osakkeiden hinnoissa vaaditaan) sekä milloin kummastakin osakkeesta luovutaan (milloin tuotto tai häviö on riittävä parikaupankäynnin lopettamiseksi) [Song and Zhang 2013]. Optimaalisia kaavoja tälle haasteelle on luotu paljon, mutta niiden matemaattisen kompleksisuuden vuoksi en aio niitä tässä tutkimuksessa esitellä.

Havainnollistetaan parikaupankäyntiä vielä keksimäni yksinkertaisen esimerkin kautta. Oletetaan osakkeiden A ja B korreloivan toisiaan kuvan 1 mukaisesti yhden viikon ajalta.



Kuva 1. Esimerkkikuvaaja kahdesta toisiaan vahvasti korreloivasta osakkeesta A ja B ja niiden keskihinnasta.

Osakkeiden A ja B keskihinta saadaan Seppäsen ja muiden [2005] mukaan kaavalla

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n},$$

jossa  $n$  on yhteenlaskettavien alkioden lukumäärä ja  $x_i$  summattava alkio. Tällöin keskihinnaksi muodostuu

$$\begin{aligned} \bar{x} &= \frac{40 + 50 + 45 + 40 + 35 + 40 + 45 + 30 + 40 + 35 + 30 + 25 + 30 + 35}{14} \\ &= 37,142857 \dots \approx 37. \end{aligned}$$

Keskihajonta saadaan puolestaan Seppäsen ja muiden [2005] tarjoamalla kaavalla

$$s_n = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}},$$

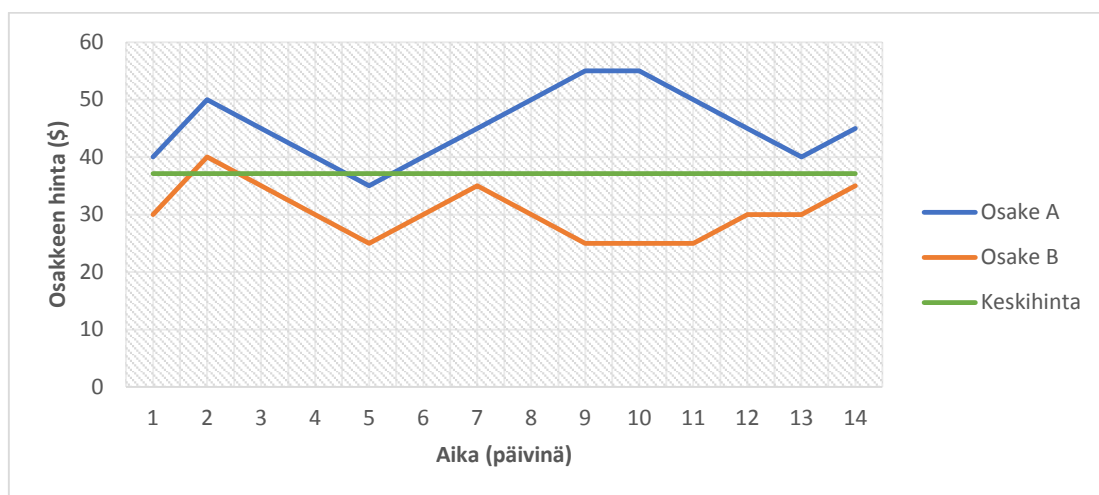
jossa  $n$  on yhteenlaskettavien alkiodien lukumäärä,  $x$  yhteenlaskettava alkio ja  $\bar{x}$  aiemmin laskemamme keskiarvo. Näin ollen osakkeiden A ja B hinnan keskihajonnaksi saadaan

$$s = \sqrt{\frac{(40 - 37)^2 + (50 - 37)^2 + (45 - 37)^2 + (40 - 37)^2 + (35 - 37)^2 + (40 - 37)^2 + (45 - 37)^2 + (30 - 37)^2 + (40 - 37)^2 + (35 - 37)^2 + (30 - 37)^2 + (25 - 37)^2 + (30 - 37)^2 + (35 - 37)^2}{14}}$$

$$= 6,740072... \approx 7$$

Zengin ja Leen [2014] kolmen askeleen mukaan parikaupankäynti on syytä aloittaa, kun suhde poikkeaa keskiarvosta kahdella keskihajonnalla. Oletetaan osakkeiden A ja B kursien jatkuvan seuraavalla viikolla (kahdeksannesta päivästä alkaen) keksimäni kuvaajan mukaisesti kuvassa 2.

Aiempia kaavoja hyödyntäen huomaamme, että yhdeksäntenä päivänä keskihajonta poikkeaa yli kahden verran keskiarvosta. Tällöin siirrytään Zengin ja Leen [2014] toiseen askeleeseen ja algoritmi käynnistää parikaupankäynnin ylimyymällä osaketta A ja ostamalla osaketta B ennalta määrättyllä hinnalla. Folger mainitsee kaupankävijöiden käyttävän usein niin sanottua "dollar-neutral" lähestymistapaa, jonka mukaan molempia osakkeita ostetaan yhtä suurilla summilla.



Kuva 2. Osakkeiden A ja B kehitys seuraavalla viikolla (kahdeksannesta päivästä alkaen).

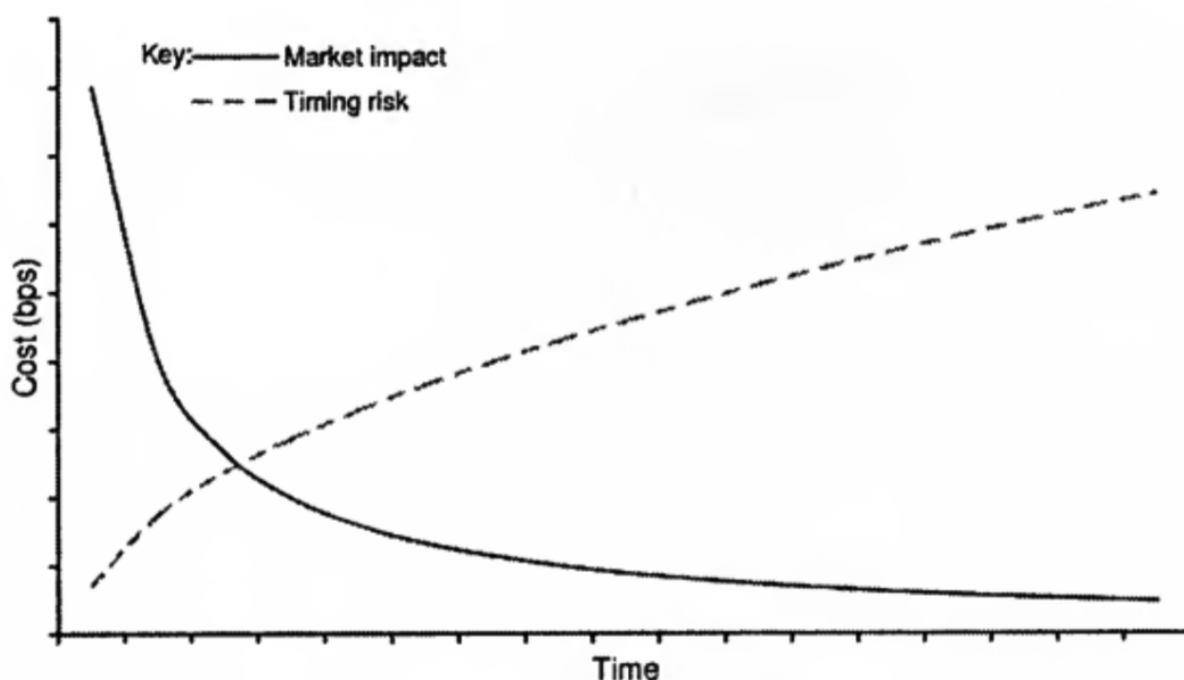
Oletetaan siis, että algoritmi ylimyy osaketta A 5 kappaletta (55\$ per osake) ja ostaa osaketta B 11 kappaletta (25\$ per osake), jolloin osakkeista A saadaan 275\$ ja osakkeisiin B menee 275\$. Zengin ja Leen [2014] kolmanteen askeleeseen siirrytään, kun osakkeiden hin-

tojen välinen suhde on palannut takaisin ennalta määrätylle tasolle, eli yleensä lähelle keskiarvoa. Näin voidaan olettaa tapahtuvan kuvan 2 kolmantenatoista päivänä. Tuolloin ostetaan 5 kappaletta osaketta A, jotka ylimyytiin parikaupankäynnin toisessa vaiheessa. Kolmantenatoista päivänä osakkeen A arvo on 40\$, eli ostoon menee 200\$. Tuottoa osakkeen A ylimyyntistä kertyi siis 75\$. Tämän lisäksi myydään toisessa vaiheessa ostamamme 11 kappaletta osaketta B, joiden kurssi on kohonnut takaisin 30\$. Näin ollen osakkeista B on kertynyt tuottoa 55\$, eli parikaupankäynti teki yhteensä 130\$ voittoa kaupankäyntikuluja huomioonottamatta.

## 5. Kustannusvetoiset algoritmit

Kuten aiemmissakin tässä tutkimuksessa esitetyissä algoritmeissa, myös *kustannusvetoisten algoritmien* (cost-driven algorithms) nimi enteilee algoritmien käyttötarkoitusta. Kustannusvetoisissa algoritmeissa tavoitteena on minimoida kaupankäynnistä syntyvät kokonaiskustannukset. Näiden lisäksi epäsuorat kustannukset kuten vaikutukset markkinoihin sekä *ajotusriskit* (timing risks) ovat tärkeitä osia kokonaiskustannuksen syntymisessä. Markkinoihin syntyvien vaikutusten minimoimista käsiteltiin vaikutusvetoisten algoritmien yhteydessä. Toimeksiannon pilkkominen pienempiin toimeksiantoihin altistaa toimeksiannot (etenkin epävakaiden osakkeiden kohdalla) ajoitusriskeille, jonka vuoksi kustannusvetoisten algoritmien on otettava ajoituksesta syntyvät riskit erityiseen huomioon. [Johnson 2010.] Ajoitusriskeillä tarkoitetaan Investopedian mukaan riskiä, jonka kaupankävijä ottaa pyrkinessään käydä kauppaa tulevaisuuden hintaennusteiden perusteella. Käytännössä tämä tarkoittaa sitä, että ostetaan osaketta liian kalliilla tai myydään liian halvalla, koska ajoitus on mennyt pieleen.

Johnson [2010] kuvailee kokonaiskustannusten minimoimista vaativaksi prosessiksi, sillä liian aggressiivinen kaupankäynti voi johtaa liian suuriin vaikutuksiin markkinoilla, kun taas liian passiivinen kaupankäynti voi johtaa ajoitusriskeihin. Kuvassa 3 olevat käyrät havainnollistavat näiden kahden tekijän suhdetta toisiinsa. Mitä suuremmalla hinnalla ja lyhyemmässä ajassa osakkeita ostetaan, sen suuremmat ovat myös vaikutukset markkinoihin. Kun taas aika on lyhyt ja summa on pieni, ajoitusriskit ovat matalat. Näiden lisäksi kaupankävijän kiireellisyys ja riskinsietokyky ovat osatekijöinä sopivan tasapainon löytämisessä kaupankäyntikulujen minimoimiseksi.



Kuva 3. Ajan ja hinnan merkitys markkinavaikutuksessa sekä ajoitusriskissä [Johnson 2010].

Ensimmäiset kustannusvetoiset algoritmit kehittyivät vaikutusvetoisten algoritmien seurauksena, kun vaikutusvetoisten algoritmien käyttäytymiseen liitettiin ajoitusriski. Kustannusvetoisia algoritmeja on kuitenkin kehitetty yhä enenevässä määrin monimutkaisten markkinamallien pohjalta arvioimaan mahdollisia kokonaiskustannuksia. Näiden arvioiden mukaan kyetään päättämään optimaalisin kaupankäyntistrategia jokaiselle toimeksiannolle. [Johnson 2010.]

*Toteutusvaje* (implementation shortfall) on kaupankäyntistrategia, jolla pyritään kuvaamaan teoreettisen osakesalkun ja toteutuneen osakesalkun tuottojen erotusta. Toisin sanoen toteutusvaje siis mittaa hintaeroa lopullisen kaupankäyntihinnan ja kaupankävijän ajattelun hinnan välillä. Toteutusvajeen laskemisessa käytetään kolmea tekijää: *paperisalkkua* (paper portfolio), *todellista salkkua* (actual portfolio) ja niin sanottua *jänissalkkua* (rabbit portfolio). Paperisalkku kuvastaa kaupankäynnin ideaalitulannetta, jossa kaikki osakkeet on vaihdettu vertailuhinnalla eikä kaupankäynnistä synny muita kuluja tai seurauksia. Todellinen salkku kuvastaa ”oikeaa elämää”, jossa osakkeet vaihdetaan oikeilla markkinoilla ja kaupankäynnistä syntyy lisäkuluja sekä seurauksia. Jänissalkun avulla puolestaan kuvataan oletettuja kaupankäyntikuluja, jossa osakkeet vaihdetaan oletetuilla markkinoilla ja kaupankäynnistä syntyvät muut kulut asettuvat paperisalkun ja todellisen salkun väliin. [Kim 2007.]

Toteutusvaje-algoritmin pääasiallinen tarkoitus on saavuttaa toimeenpanossa hinta, joka minimoi kaupankävijän ajatellun hinnan ja toimeenpanon hinnan välisen vajeen. Jotta tällaiseen lopputulokseen päästäisiin, algoritmin on toimittava juuri oikeaan aikaan tasapainotellakseen markkinavaikutusten ja ajoitusriskin välillä (ks. kuva 3). Tämä tarkoittaa usein sitä, että algoritmin on toimittava vain sen aikaa kuin on pakko. [Johnson 2010.]

Havainnollistetaan prosentuaalisen toteutusvajeen  $X$  syntymistä hieman mukauttamalla Smagan [2014] antamaa kaavaa, joka perustuu CFA (Chartered Financial Analyst) instituutin kolmannen tason ohjelmaan. Kaavaksi muodostuu

$$X = \frac{P_G - (R_G - C)}{P_i},$$

jossa  $P_G$  on paperisalkun tuotto,  $R_G$  on todellisen salkun voitto,  $C$  kuvastaa kaupankäynnistä syntyneitä kustannuksia ja  $P_i$  vastaavasti paperisalkun sijoitusta.

Hyödynnetään esitettyä kaavaa vajeen syntymisestä Smagan [2014] tarjoamassa esimerkissä. Oletetaan, että osaketta A ostetaan 10 kappaletta hintaan 20\$ per osake, eli yhteensä 200\$:lla. Päivän päätteeksi osakkeen hinta on 20,20\$, eli omistettujen osakkeiden arvo on 202\$ ja tuottoa paperisalkkuun on kertynyt 2\$. Todellisessa salkussa tilanne ei ole yhtä yksinkertainen. Osaketta A saadaankin ostettua vain 8 kappaletta hintaan 20,15\$ per osake, eli yhteensä 161,20\$:lla. Kaupankäynnin sulkeuduttua osakkeen arvo on noussut 20,20\$, jolloin omistettujen osakkeiden arvo on 161,60\$, joten tuottoa kertyi 0,40\$. Välityspalkkiota menee 0,03\$ per osake (yhteensä 0,24\$) ja muihin kuluihin 0,02\$ per osake (yhteensä 0,16\$). Kritzmanin ja muiden [2006] antamaa kaavaa hyödyntäen prosentuaaliseksi toteutusvajeeksi saadaan siis

$$\frac{2 - (0,40 - 0,24 - 0,16)}{200} = 1,0\%,$$

jolloin potentiaalisesta sijoituksesta menetettiin yksi prosentti.

Toteutusvaje-algoritmillla siis pyritään tuomaan ideaalitilanne ja realiteetti mahdollisimman lähelle toisiaan, jotta tuotot saataisiin maksimoitua. Kuten monet muutkin kaupankäyntialgoritmit, myös toteutusvaje-algoritmit ovat suunniteltu nykyään monimutkaisten markkinamallien ympärille, joiden avulla kyetään ennustamaan kaupankäynnistä syntyviä kuluja ja näin ollen toimimaan järkevästi niiden minimoimiseksi. Osa toteutusvaje-algoritmeista etsii optimaalista kaupankäyntistrategiaa tarkkailemalla jatkuvasti markkinaolosuhteita ja suorittamalla yksityiskohtaista kustannusanalyysia ennen jokaisen toimeksiannon asettamista. Mitä kehittyneempi algoritmi on, sen vaikeammissa markkinaolosuhteissa se pärjää. Kovin suuria eroja algoritmien välillä ei kuitenkaan ole vielä havaittavissa, sillä yleisesti ottaen ne käyttävät hyvin samankaltaisia kaupankäyntimalleja. [Johnson 2010.]

## 6. Yhteenveto

Tutkimuksessa käsiteltiin algoritmista kaupankäyntiä yleisellä tasolla, sekä paneuduttiin kolmeen erilaiseen algoritmikategoriaan, joista kustakin esiteltiin yksi algoritmi. Ensimmäinen kategoria koostui vaikutusvetoisista algoritmeista, jotka keskittyvät minimoimaan markkinoihin syntyvät vaikutukset käydessään osakekauppaa. Kategoriasta esiteltiin volyympainotteinen keskihinta, joka pyrkii toteuttamaan algoritmeille tyypillistä piirrettä pilkkomalla toimeksiannon useaksi pieneksi toimeksiannoksi. Toimeksiannon koko riippuu osakkeen kaupankäyntihistoriasta, sillä algoritmi tarkastelee osakkeen kaupankäyntivolyyymia ja pilkkoo toimeksiannon sen perusteella sopivan kokoisiksi toimeksiannoiksi. Koen vaikutusvetoisissa algoritmeissa eduksi niiden perusperiaatteen tehdä mahdollisimman vähän vaikutusta markkinoihin, sillä tämä edesauttaa markkinoita pysymään vakaina ja jollain tapaa ennustettavina, mikä puolestaan helpottaa niin ihmisten kuin algoritmienkin kaupankäyntiä. Vaikutusvetoisten algoritmien heikkoutena näen puolestaan niiden toimintaperiaatteen rajoittuneisuuden, sillä niiden tehtävänä on vain toteuttaa toimeksianto annetussa aikamääreessä riippumatta markkinaolosuhteista. Mikäli markkinaolosuhteet muuttuvat radikaalisti algoritmin toiminnan aikana, algoritmin toiminta voi olla kohtalokasta.

Toinen käsitelty kategoria sisälsi tietoa opportunistisista algoritmeista. Nämä poikkeavat vaikutusvetoisista algoritmeista siinä, että niiden pääpaino on nimenomaan markkinaolosuhteiden tarkkailussa ja sieltä löydettyissä tuottomahdollisuuksissa. Tähän algoritmikategoriaan liittyy vahvasti parikaupankäynti, minkä ideana on etsiä kaksi toisiaan vahvasti korreloivaa osaketta ja hyötyä tilanteesta, jossa näiden kahden välinen korrelaatio heikkenee hetkeksi. Opportunistisissa algoritmeissa etuna näen niiden voiton tuottamisen markkinaolosuhteista riippumatta. Ne eivät vaadi markkinoiden laskua tai kasvua jotta tuottoa saadaan tehtyä, vaan ne pystyvät hyötymään markkinoista oli tilanne mikä tahansa. Toisaalta tällä asialla on myös varjopuolensa ja sitä voidaan tarkastella eettisestä näkökulmasta: onko oikein kerätä itselle hyötyä toisen vaikean tilanteen kustannuksella? Toki osakkeiden kurssit muuttuvat jatkuvasti eikä niiden hetkellisille poikkeamille ole välttämättä mitään oikeaa syytä, mutta selkeään korrelaation heikkenemiseen on usein taustalla jokin järkevä selitys.

Viimeinen algoritmikategoria oli kustannusvetoiset algoritmit, joiden pääasiallinen tarkoitus oli käydä osakekauppaa mahdollisimman kustannustehokkaasti, eli minimoida kaupankäynnistä syntyvät kokonaiskustannukset. Kategoriasta käsiteltiin toteutusvaje-strategia, joka kuvastaa mahdollisten tuottojen ja realisoitujen tuottojen välistä erotusta. Koen kustannusvetoisten algoritmien heikkoudeksi niiden monimutkaisuuden ja riippuvuuden moniin tekijöihin. Kaupankäynnistä syntyviin kokonaiskustannuksiin vaikuttaa useat tekijät, jonka vuoksi algoritmin on oltava hyvin toteutettu ja varmatoiminen toimiakseen tuottoisasti. Toisaalta myös koen, että tämä voidaan kääntää kustannusvetoisten algoritmien

eduksi. Algoritmi, joka huomioi monet tekijät, on oikein toimiessaan todennäköisesti tuottavampi kuin algoritmi, joka keskittyy vain toteuttamaan toimeksiannon ottamatta huomioon mitä markkinoilla oikeasti tapahtuu.

Se, mitä algoritmisen kaupankäynti tuo tulevaisuudessa tullessaan on hyvin haastava ennustaa. Kuten kohdassa 2.3 mainitsin, algoritmisessa kaupankäynnissä käytettävät algoritmit eivät ole enää niinkään riippuvaisia tietokoneiden laskentatehosta, vaan ihmisten käyttäytyminen ja tekoälyn sekä neuroverkkojen kehittyminen nousevat entistä suurempaan rooliin. Näiden lisäksi uskon myös algoritmisen kaupankäynnin kehittymisen johtavan uusiin säännöksiin ja valvonnan tiukentumiseen, mikä omalta osaltaan hankaloittaa tulevaisuuden näkymien ennustamista.

## Viiteluettelo

- Stephen A. Berkowitz, Dennis E. Logue and Eugene A. Noser. 1988. The total cost of transactions on the NYSE. *The Journal of Finance* 43, 1, 97–112.
- Danny Busch. 2016. MiFID II: regulating high frequency trading, other forms of algorithmic trading and direct electronic market access. *Law and Financial Markets Review* 10, 2, 72–82.
- Robert J. Elliot, John Van Der Hoek and William P. Malcolm. 2005. Pairs trading. *Quantitative Finance* 5, 3, 271–276.
- Joseph Engelberg, Pengjie Gao and Ravi Jagannathan. 2008. An anatomy of pairs trading: The role of idiosyncratic news, common information and liquidity. In: *Proc. of the Third Singapore International Conference on Finance 2009*.
- Jean Folger. Pairs Trade Example. Investopedia. <<http://www.investopedia.com/university/guide-pairs-trading/pairs-trade-example.asp>> (Checked 7.11.2017).
- Christopher Frei and Nicholas Westray. 2013. Optimal execution of VWAP order: A stochastic control approach. *Mathematical Finance* 25, 3, 612–639.
- Investopedia. Timing risk. <<https://www.investopedia.com/terms/t/timingrisk.asp>> (Checked 13.11.2017).
- Barry Johnson. 2010. *Algorithmic Trading and DMA*. 4Myeloma Press London.
- Kendall Kim. 2007. *Electronic and Algorithmic Trading Technology: The Complete Guide*. Elsevier Science.
- Andrei A. Kirilenko and Andrew W. Lo. 2013. Moore's law versus Murphy's law: Algorithmic trading and its discontents. *The Journal of Economic Perspectives* 27, 2, 51–72.
- Mauricio Labadie and Charles-Albert Lehalle. Optimal algorithmic trading and market microstructure. 2010. <<https://hal.archives-ouvertes.fr/hal-00590283v2/document>>. (Checked 6.10.2017).



- Peter Lattman. 2010. Former Goldman Programmer Found Guilty of Code Theft. The New York Times. <<https://dealbook.nytimes.com/2010/12/10/ex-goldman-programmer-is-convicted/>> (Checked 21.10.2017).
- Marc Lenglet. 2011. Conflicting codes and codings: How algorithmic trading is reshaping financial regulation. *Theory, Culture & Society* 28, 6, 44–66.
- Edward A. Leshik and Jane Cralle. 2011. *An Introduction to Algorithmic Trading: Basic to Advanced Strategies*. Wiley.
- Handong Li and Xunyu Ye. 2013. A dynamic, volume-weighted average price approach based on the fast Fourier transform algorithm. *Asia-Pacific Journal of Financial Studies* 42, 6, 969–991.
- Lauren Liebenberg. 2002. *The Electronic Financial Markets of the Future: Survival Strategies of the Broker-Dealer*. Palgrave Macmillan.
- Chris A. Mack. 2011. Fifty Years of Moore’s Law. *IEEE Transaction on Semiconductor Manufacturing* 24, 2, 202–207.
- Jerry W. Markham and Daniel J. Harty. 2008. For whom the bell tolls: The demise of exchange trading floors and the growth of ECNs. *Journal of Corporation Law* 33, 4, 865–939.
- Raimo Seppänen, Matti Kervinen, Irma Parkkila, Lea Karkela ja Pekka Meriläinen. 2005. *Maol-taulukot*. Otavan Kirjapaino Oy.
- Jérémie Smaga. 2014. CFA Level III: Implementation shortfall. <<http://blog.smaga.ch/cfa-level-iii-implementation-shortfall>> (Checked 14.11.2017).
- Qingshuo Song and Qing Zhang. 2013. An optimal pairs-trading rule. *Automatica* 49, 10, 3007–3014.
- Suomen Pörssisäätiö. 2017a. *Osakeopas*. <<http://www.porssisaatio.fi/wp-content/uploads/2017/06/Osakeopas-2017.pdf>> (Käytetty 18.10.2017).
- Suomen Pörssisäätiö. 2017b. *Ylimyynti*. <<http://www.porssisaatio.fi/blog/dictionary/ylimyynti/>> (Käytetty 23.11.2017).
- Zhengqin Zeng and Chi-Guhn Lee. 2014. Pairs trading: Optimal thresholds and profitability. *Quantitative Finance* 14, 11, 1881–1893.

# IT-hankintojen ongelmat julkisella sektorilla

**Santeri Tuomisto**

## Tiivistelmä

Tässä tutkielmassa tarkastellaan julkisen sektorin järjestelmähankintojen ongelmia. Tutkielmassa selviää, että hankintojen ongelmat ovat yleensä monimutkaisia, eikä niihin löydy yhtä selvää ratkaisua. Suuria ongelmia hankinnoissa ovat esimerkiksi kommunikaatio-ongelmat, puutteelliset taidot ja vanhentuneet projektinhallintamenetelmät. Ratkaisuksi näihin voisi esimerkiksi olla konsultaatio toisten hankintayksiköiden kanssa sekä ketterien menetelmien hyödyntäminen.

Hankintojen ongelmia esitellään kahden esimerkkitapauksen avulla. Esimerkkitapauksina työssä ovat VR:n lippujärjestelmä ja Poliisin Vitja-järjestelmä. Esimerkkitapausten läpikäynnissä lähdeaineistona toimii pääasiassa medialähteet ja toimijoiden itsensä julkaisemat raportit. Ongelmien läpikäynnissä aineistona käytetään tutkimuskirjallisuutta.

Hankintaorganisaatiolla tulisi olla ymmärrys organisaationsa tarpeista ja kokonaiskuva tarvittavista järjestelmistä, jotta hankinnoissa voitaisiin onnistua. Ilman selvää hankintastrategiaa hankinnat harvoin onnistuvat. Pahimmassa tapauksessa hankitaan yhteensopimattomia palasia, joiden käytettävyys on huono. Tärkeää olisi ymmärtää, että järjestelmähankinta ei ole koskaan valmis, vaan se on jatkuva prosessi.

Aihe on ajankohtainen ja kiinnostava, koska järjestelmähankinnoissa on selvästi ongelmia ja moni hanke epäonnistuu. Epäonnistuneisiin hankintoihin kuluu veroeuroja ja aikaa. Huonosti toimivat järjestelmät taas vaikeuttavat ihmisten työntekoa.

**Avainsanat ja -sanonnat:** julkiset IT-hankinnat, tietojärjestelmä, järjestelmähankinta, hankintalaki, ketterät menetelmät, toimittajalukko, avoin lähdekoodi

## 1. Johdanto

Mediassa törmää usein uutisiin, jotka liittyvät julkisen sektorin tietojärjestelmä-hankintoihin. Yleensä uutiset liittyvät hankintojen ongelmiin. Miksi järjestelmän hankkiminen on niin vaikeaa? Kuinka usein niissä epäonnistutaan? Mitä ongelmia niihin liittyy? Epäonnistutaanko julkisella sektorilla yksityistä sektoria enemmän? Muun muassa näihin kysymyksiin pyrin vastaamaan tutkielmassani. Aihe on ajankohtainen ja kiinnostava, sillä epäonnistuneisiin hankintoihin kuluu aikaa ja verorahoja.

Julkisella sektorilla hankinnat tehdään hankintalain pohjalta, jonka vuoksi johdannon jälkeen luvussa 2 käsitellään hankintalakia. Luvussa 3 tutkitaan julkisia IT-hankintoja Suomessa. Luvussa käydään läpi esimerkiksi suurimpia toimittajia ja kuinka usein hankinnat epäonnistuvat. Luvussa 4 esitellään kaksi esimerkkitapausta julkiselta sektorilta, jotka kohtasivat ongelmia hankintaprosessissa. Esimerkkitapausten läpikäynnissä lähdeaineistona toimii pääasiassa medialähteet ja toimijoiden itsensä julkaisemat raportit. Luvussa 5 käsitellään varsinaisia ongelmia, joita hankintoihin liittyy. Luvussa pohditaan myös hankintalain roolia, sekä käydään läpi esimerkkitapausten ongelmakohtia. Luvussa 6 tarjotaan nelikentän muodossa eräänlaista ratkaisumallia ohjenuoraksi julkisiin hankintoihin. Luvussa 7 on tutkielman yhteenveto.

## 2. Julkiset hankinnat, hankintalaki ja JIT 2015 -ehdot

Julkinen hankinta on hankinta, jonka tekijöinä ovat laissa tarkoitetut hankintayksiköt. Hankintalain hankintayksiköt ovat [Julkisten hankintojen neuvontayksikkö 2017a.]:

1. valtion, kuntien ja kuntayhtymien viranomaiset
2. evankelis-luterilainen kirkko ja ortodoksinen kirkko sekä niiden seurakunnat ja muut viranomaiset
3. valtion liikelaitokset
4. julkisoikeudelliset laitokset
5. mikä tahansa hankinnan tekijä silloin, kun se on saanut hankinnan tekemistä varten tukea yli puolet hankinnan arvosta kohdissa 1 – 4 tarkoitetulta hankintayksiköltä.

Suomen julkisen sektorin ulkoisten hankintojen kokonaisarvo vuonna 2013 oli noin 34 miljardia euroa, joka vastaa noin 18% Suomen bruttokansantuotteesta. Tästä kokonaisarvosta noin 20 miljardia kuuluu julkisten hankintojen lainsäädännön piiriin. [Eskola *et al.* 2017, s. 19.]

## 2.1. Hankintalaki

Hankintalakia sovelletaan julkisiin hankintoihin, kun hankinnan arvo ylittää joko EU-kynnysarvon tai kansallisen kynnysarvon. Kansallisen kynnysarvon alle jääviin hankintoihin ei sovelleta hankintalakia. Uusi hankintalaki astui voimaan Suomessa 1.1.2017 ja sen lainsäädäntö perustuu EU:n julkisia hankintoja koskeviin direktiiveihin. Hankintalaissa ja EU-lainsäädännössä määritetyt kynnysarvot löytyvät muun muassa hankinnat.fi-verkkosivustolta, jota ylläpitää Julkisten hankintojen neuvontayksikkö. EU-kynnysarvot palveluhankinnoissa ovat 1.1.2016 alkaen valtion keskushallintoviranomaisille 135 000 euroa. Muille hankintayksiköille, muun muassa kunnille, EU-kynnysarvoraja palveluhankinnoissa on 209 000 euroa. Erityisaloille, kuten esimerkiksi sosiaali- ja terveystalveuille, sovelletaan omia kynnysarvoja. Kansalliset kynnysarvot puolestaan ovat 1.1.2017 alkaen palveluille 60 000 euroa. Sosiaali- ja terveystalvelujen hankintojen kansallinen kynnysarvo on 400 000 euroa. [Julkisten hankintojen neuvontayksikkö 2017b.]

Hankintalain tavoitteeksi määritellään julkisten varojen käytön tehostaminen sekä laadukkaiden, innovatiivisten ja kestävien hankintojen edistäminen. Lain tavoitteena on myös turvata kaikille tasapuoliset osallistumismahdollisuudet julkisiin hankintoihin. [Eskola *et al.* 2017, s. 23.]

Hankintalaissa on säädetty julkisten hankintojen periaatteiksi tasapuolinen ja syrjimätön kohtelu, avoimuus ja suhteellisuus. Kaikissa hankinnan vaiheissa on noudatettava näitä periaatteita. Avoimuus perustuu oikeuskäytäntöön ja hallinnon avoimuutta koskeviin vaatimuksiin. Hankintalaissa avoimuus näkyy esimerkiksi siten, että hankinnoista on ilmoitettava julkisesti ja hankintaan liittyvien asiakirjojen tulee olla julkisia. Suhteellisuudella tarkoitetaan, että hankinnan vaatimusten tulee olla oikeassa suhteessa tavoiteltavan päämäärän kanssa. Tasapuolinen ja syrjimätön kohtelu kattaa kaiken syrjinnän. [Eskola *et al.* 2017, ss. 34-36.]

Oli kyseessä sitten EU-hankinta tai kansallinen hankinta, molemmissa tapauksissa tarjouksista on valittava kokonaistaloudellisesti edullisin tarjous. Kokonaistaloudellisesti edullinen tarjous on joko hinnaltaan halvin, kustannuksiltaan edullisin tai paras hinta-laatusuhteeltaan. Mikäli EU-hankinnassa käytetään valintaperusteena ainoastaan halvinta hintaa, on hankkijan perusteltava valinta. Tämä perusteluvollisuus ei koske kansallisia hankintoja, joiden perusteena on halvin hinta. [Julkisten hankintojen neuvontayksikkö 2017c.]

## 2.2. JIT 2015 -ehdot

Hankintalaki ulottuu myös IT-hankintoihin. JIT 2015 -ehdot, eli julkisten IT-hankintojen ehdot, ovat Julkisen hallinnon tietohallinnon neuvottelukunnan laatimat sopimusehdot, joiden käyttämistä suositellaan julkishallinnon yksiköille julkisissa IT-hankinnoissa. Ehdot ovat päivitetty versio vanhemmista JIT 2007 -ehdoista. Hankintayksiköillä ei ole velvollisuutta käyttää JIT-ehtoja, vaan ne ovat laadittu hankintayksiköille avuksi, ettei jokaisen IT-hankinnan kohdalla tarvitsi kirjoittaa uusia sopimusehtoja. Hankintayksikkö saa halutessaan poiketa ehdoista tai soveltaa omia ehtoja. [Järvenoja *et al.* 2015, ss. 21-24.]

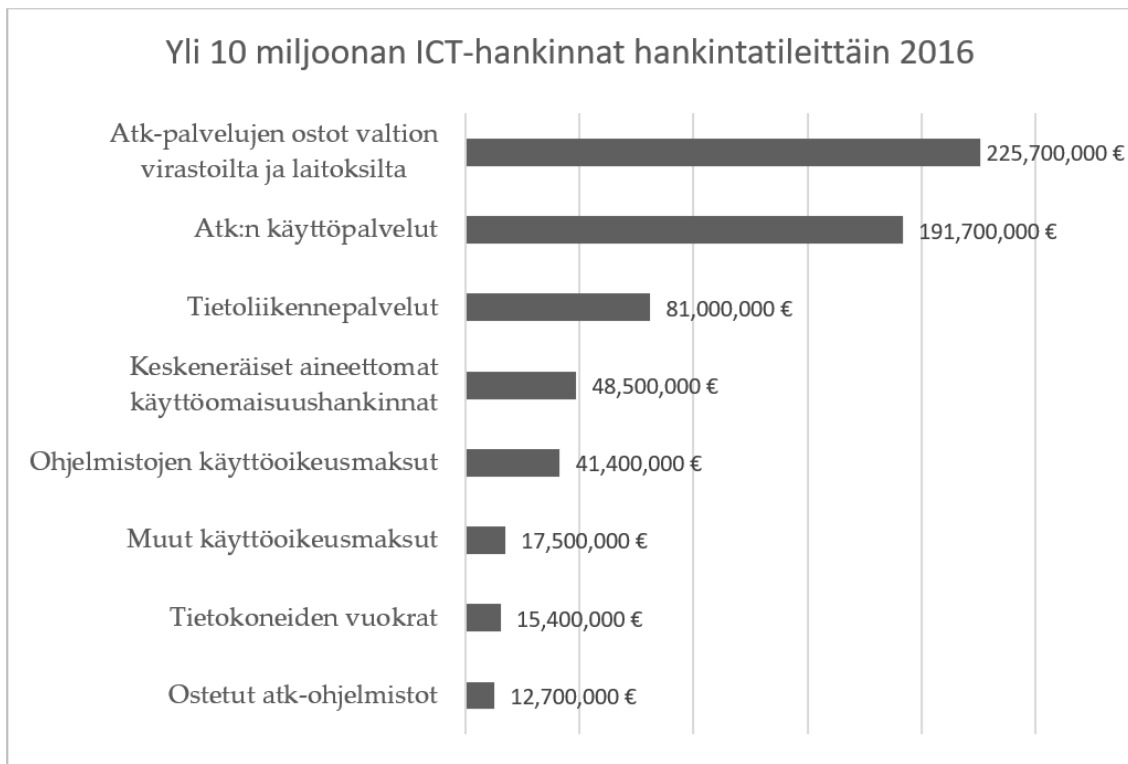
Uudistettujen JIT 2015 -ehtojen tavoitteena oli sopimusten osalta mahdollistaa kilpailukykyisten tarjousten tekeminen ja hankintojen onnistuminen. Ehdossa esimerkiksi pyritään ehkäisemään toimittajalukkojen syntymistä. Toimittajalukolla tarkoitetaan tilannetta, jossa asiakas on jumissa tietyn toimittajan kanssa. Toimittajalukossa toimittajan vaihtamisesta seuraisi merkittäviä kustannuksia tai haittaa toiminnalle. JIT 2015 -ehdot tarjoavat myös sopimusehdot järjestelmien hankkimiseen avointa lähdekoodia ja ketteriä menetelmiä hyödyntäen. [Järvenoja *et al.* 2015, ss. 24-38.]

## 3. Julkiset IT-hankinnat Suomessa

Tutkihankintoja.fi -sivuston kautta valtio tarjoaa tietoa siitä mitä hankitaan, mistä hankitaan ja minne hankitaan. Vuonna 2016 tuotteita ja palveluita hankittiin yhteensä noin 4 miljardilla eurolla. Tästä summasta ICT-hankintojen määrä oli noin 650 miljoonaa euroa. Vuonna 2016 hankituista tuotteista tai palveluista 16% oli siis ICT-hankintoja. ICT-hankinnat kattavat hankitut tavarat, palvelut sekä ohjelmistot. [Tutkihankintoja.fi 2017.]

Vuoden 2016 yli 10 miljoonan euron ICT-hankinnat on jaoteltu tarkemmin hankintatileittäin kuvassa 1. Suurin hankintatili oli atk-palvelujen ostot valtion virastoilta ja laitoksilta, joka oli 225 700 000 euroa. Tästä 178 800 000 euroa toimitti Valtion tieto- ja viestintätekniikkakeskus Valtori. Toiseksi suurin hankintatili oli Atk:n käyttöpalvelut, joka oli 191 700 000 euroa. Tästä 39 700 000 euroa toimitti Tieto Finland Oy. [Tutkihankintoja.fi 2017.]

Suurimmat ICT-toimittajat vuonna 2016 (kuva 2) olivat edellä mainitut Valtion tieto- ja viestintätekniikkakeskus Valtori, jolta hankittiin 195 800 000 eurolla, sekä Tieto Finland Oy, jolta hankittiin 43 300 000 eurolla. Näiden lisäksi Crayon Oy:lta hankittiin yhteensä 23 900 000 euron edestä, joista 20 200 000 euroa oli ohjelmistojen käyttöoikeusmaksuja. [Tutkihankintoja.fi 2017.]



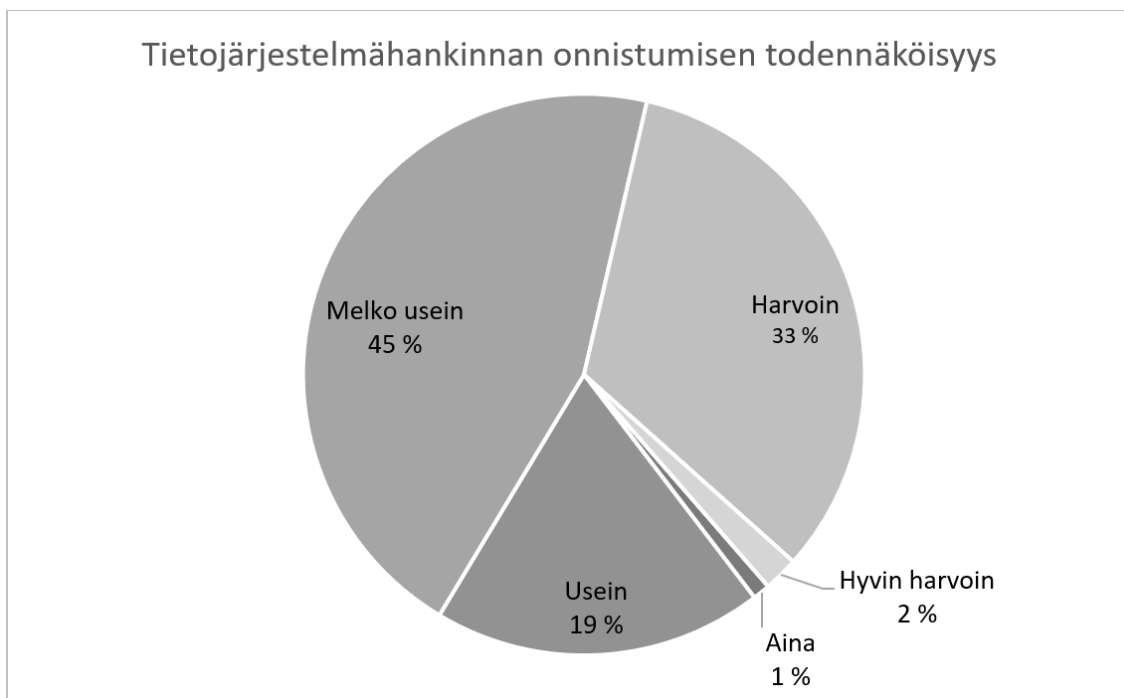
Kuva 1. Yli 10 miljoonan ICT-hankinnat 2016 [tutkihankintoja.fi 2017].



Kuva 2. ICT-hankkeiden suurimmat toimittajat 2016 [tutkihankintoja.fi 2017].

Standish Groupin [2016] tekemässä CHAOS-raportissa 29% ohjelmistoprojekteista onnistui ajallaan ja pysyi budjetissa, kun taas haasteita oli 52% projekteista ja jopa 19% projekteista epäonnistui. Raportissa tutkittiin 50 000 projektia ympäri maapallon. Suomessa Celkee Oy, Tietotekniikan liitto ry ja Ohjelmistoyrittäjät ry ovat yhteistyössä toteuttaneet Tietojärjestelmien hankinta Suomessa

2013 -tutkimuksen [Celkee Oy *et al.* 2013]. Tutkimukseen osallistui 104 vastaajaa, joista 67 edusti tietojärjestelmien tilaajaorganisaatiota ja 37 toimittajaorganisaatiota. Vastaajista 25% oli julkiselta sektorilta. Kun kysyttiin tietojärjestelmähankinnan onnistumisen todennäköisyyttä (kuva 3), jopa 35% koki onnistuvansa vain harvoin tai hyvin harvoin. Suurin osa eli 45% koki onnistuvansa melko usein, 19% onnistuvansa usein, ja vain 1% aina. Noin kolmasosa vastaajista siis kokee epäonnistuvansa useammin järjestelmähankinnassa kuin onnistuvansa. Tutkimuksessa tätä pidettiin hälyttävänä määränä, sillä tietojärjestelmähankinnat ovat isoja ja kalliita projekteja, jotka vaikuttavat koko organisaation toimintaan. Vastauksissa ei ollut eroja julkisen ja yksityisen puolen välillä. [Celkee Oy *et al.* 2013, ss. 19-20.]



Kuva 3. Koettu tietojärjestelmähankinnan onnistumisen todennäköisyys tilaajanäkökulmasta [Celkee Oy *et al.* 2013, s. 20].

Vuonna 2016 Sofigaten, TIVIAN ja Aalto-yliopiston vuosittain tekemässä Tietohallintojen johtaminen Suomessa -tutkimuksen 174 vastaajasta 24% oli julkiselta sektorilta. Vastaukset ovat kuitenkin eroteltu erikseen vastaajan organisaation perusteella, joten julkisen sektorin vertaaminen yksityiseen sektoriin onnistuu. Kysymykseen kuinka usein yrityksenne kehityshankkeet pysyvät niille asetetussa aikataulussa ja budjetissa (taulukko 1), julkisen sektorin vastaajista 3% vastasi aina, 65% melko usein, 30% harvoin, 0% ei koskaan ja 2% ei osannut sa-

noa. Kaikista vastaajista 34% ilmoitti, että projektit pysyvät aikataulussa ja budjetissa harvoin tai ei koskaan. [Sofigate *et al.* 2016, s. 27.] Tulokset ovat samansuuntaisia kuin Tietojärjestelmien hankinta Suomessa 2013 -tutkimuksessa saadut tietojärjestelmähankinnan onnistumisen todennäköisyydet (kuva 3) [Celkee Oy *et al.* 2013, s. 20].

Taulukko 1. Kuinka usein kehityshankkeet pysyvät niille asetetussa aikataulussa ja budjetissa [Sofigate *et al.* 2016, s. 27].

	Aina	Melko usein	Harvoin	Ei koskaan	En osaa sanoa
ICT-toimiala	0%	50%	43%	0%	7%
Teollisuus	2%	62%	28%	4%	4%
Kauppa	0%	73%	27%	0%	0%
Palvelu	0%	56%	31%	6%	7%
<u>Julkinen sektori</u>	3%	65%	30%	0%	2%
<b>Yhteensä</b>	<b>1%</b>	<b>61%</b>	<b>32%</b>	<b>2%</b>	<b>4%</b>

Julkisen sektorin vastaukset ovat samassa linjassa yleisen keskiarvon kanssa, eikä suurempia heittoja ole. Toimialoista teollisuus ja julkinen sektori ovat ainoita, jotka kertovat projekteista, jotka pysyvät aina aikataulussa ja budjetissa. [Sofigate *et al.* 2016, s. 27.]

Tietohallintojen johtaminen Suomessa -tutkimuksessa kysyttiin myös, joutuvatko organisaatiot käyttämään resursseja projektien uudelleen suunnitteluun, koska niiden alkuperäinen aikataulu tai suunnitelma ei toteutunut (taulukko 2). Julkisen sektorin vastaajista 5% ei osannut sanoa, 3% vastasi ettei lainkaan, 63% vastasi ettei taloudellisesti merkittäviä määriä varoja ja jopa 29% vastaajista raportoi joutuvansa käyttämään taloudellisesti merkittäviä määriä varoja projektien uudelleen suunnitteluun. Julkisella sektorilla joudutaan 22 prosentin keskiarvoon verrattuna käyttämään merkittävästi enemmän varoja projektien uudelleen suunnitteluun. [Sofigate *et al.* 2016, ss. 29-30.]



Taulukko 2. Kuinka paljon resursseja organisaatiot joutuvat käyttämään projektien uudelleen suunnitteluun, koska niiden alkuperäinen aikataulu tai suunnitelma ei toteutunut [Sofigate *et al.* 2016, s. 29].

	Ei lainkaan	Ei taloudellisesti merkittävästi	Taloudellisesti merkittävästi	En osaa sanoa
ICT-toimiala	7%	36%	36%	21%
Teollisuus	4%	60%	22%	14%
Kauppa	0%	91%	0%	9%
Palvelu	3%	69%	22%	6%
<u>Julkinen sektori</u>	3%	63%	29%	5%
<b>Yhteensä</b>	<b>3%</b>	<b>64%</b>	<b>22%</b>	<b>11%</b>

Yleisesti edellä mainituissa tutkimuksissa ei ollut suuria eroja julkisen ja yksityisen sektorin välillä. Näiden perusteella Suomessa noin 35% hankinnoista epäonnistuu, 34% ei pysy aikataulussa tai budjetissa ja 22% näistä projekteista joudutaan käyttämään taloudellisesti merkittäviä määriä varoja uudelleen suunnitteluun. Molemmilla sektoreilla siis epäonnistutaan tietojärjestelmähankinnoissa huolestuttavan paljon. Tässä työssä perehdyn kuitenkin julkiseen sektoriin ja sen tietojärjestelmähankintojen ongelmiin. Seuraavaksi käyn läpi kaksi suurempaa tietojärjestelmähankintaa julkiselta sektorilta.

## 4. Esimerkkitapaukset

Tässä luvussa esittelen kaksi esimerkkitapausta julkiselta sektorilta, joiden hankinnassa on ollut ongelmia. Esimerkkitapausten läpikäynnissä lähdeaineistona toimii pääasiassa medialähteet ja toimijoiden itsensä julkaisemat raportit. Esimerkkitapausten varsinainen ongelmien pohdinta tapahtuu luvussa 5.

### 4.1. VR:n lipunmyyntijärjestelmä

*”VR:n uusi lipunmyyntijärjestelmä maksoi 15 miljoonaa euroa. Sen rakentaminen kesti kolme vuotta. Järjestelmä kesti käytössä noin kolme tuntia.”*

[Vanhala 2012.]

VR:n historian suurimman it-projektin rakentaminen aloitettiin vuonna 2008. Verkkokaupan päivityksen lisäksi uudistuksessa vaihdettiin lippuautomaatit ja konduktöörin laitteet. Accenture Oy vastasi järjestelmän tekemisestä, kun taas sen alihankkija Enfo Oy toimitti lippuautomaatit ja konduktöörin laitteet. Tieto Oy vastasi palvelimista. Järjestelmä valmistui vuonna 2011 syyskuussa, noin puoli vuotta myöhässä alkuperäisestä aikataulusta. [Vanhala 2012.]

Julkaisuaamuna verkkokauppa meni tukkoon yleisöryntäyksen takia noin kolmen tunnin jälkeen uuden järjestelmän julkaisusta. Tekniikka&Talous uutisoi samana aamuna järjestelmän jumiutumista ja siitä, että vain päivää aikaisemmin VR oli kertonut varautuneensa kunnolla uudistukseen muun muassa palkkaamalla lisäväkeä. [Repo 2011.]

VR pyysi Tietoa kolminkertaistamaan palvelinkapasiteetin heti ensimmäisenä päivänä, kun yleisöryntäyksen jälkeen verkkokaupan ongelmat alkoivat. Myöhemmin samana päivänä myös lippuautomaatit lopettivat toimintansa. Palvelinkapasiteetin lisäyksestä ei ollut apua, sillä jälkikäteen selvisi, että järjestelmässä oli jäänyt korjaamatta vika, jonka takia palvelimet toimivat epävakaasti. Myöskin lippuautomaateissa oli vika, joka kuormitti järjestelmää. Automaatit olivat lopulta suljettuna kaksi viikkoa, kun vikaa korjattiin. [Vanhala 2012.]

Suuren kävijämäärän vuoksi otettiin käyttöön rajoitin, joka rajoittaa kävijämäärää. Virheellisesti toimineen rajoittimen vuoksi osa maksaneista asiakkaista lensi ulos järjestelmästä ennen kuin he ehtivät saada lipun. [Vanhala 2012.]

Julkaisun jälkeen käytettävyyssiantuntija Aapo Puskala kirjoitti käytettävyyssarvion VR:n verkkokaupasta. Raportissa tuotiin esille 26 kriittistä ongelmaa, 54 merkittävää ongelmaa ja 108 kohtalaista ongelmaa. Merkittäviä ongelmia oli muun muassa sivujen hitaus sekä monimutkainen 15-vaiheinen lipunostoprosessi. [Puskala 2011.]

Vuonna 2015 Helsingin Sanomat uutisoi, että VR joutuu ostamaan 2,7 miljoonan euron tietojärjestelmäuudistuksen Accenturelta, koska muut yhtiöt eivät pääse käsiksi vuonna 2011 rakennetun järjestelmän ytimeen [Saarinen 2015a]. Uudistuksessa muun muassa lisättiin ominaisuus ostaa lippuja öisin, sillä aiemmin järjestelmä sulki lippumyynnin öisin puoli kahdesta neljään asti [Saarinen 2015b].

## **4.2. Poliisin Vitja-järjestelmä**

Vitja eli Viranomaistietojärjestelmä on työn alla oleva tietojärjestelmä, joka on tarkoitettu Poliisin, Oikeusministeriön, Rajavartiolaitoksen, Tullin ja Puolustusvoimien käyttöön. Nykyiset järjestelmät ovat vanhentuneita ja hidastavat poliisi-

sien työntekoa. Hankkeen on tarkoitus koota nykyiset operatiiviset tietojärjestelmät yhteen ja tähdätä paperittomuuteen. Hankkeen alustava kustannusarvio on kokonaisuudessaan noin 25 miljoonaa euroa. [Passi 2014.]

MTV uutisoi vuonna 2015, että pitkäksi venyneiden oikeusprosessien takia Suomi on saanut Euroopan ihmisoikeustuomioistuimelta useita langettavia tuomioita. Tuolloin Poliisin ja oikeuden välillä ei käytetty sähköistä järjestelmää, vaan tieto liikkui paperilla. Vitjan olisi tarkoitus helpottaa myös oikeusprosesseja. [Auramies 2015.]

Vitja-hanke aloitettiin vuonna 2009 ja sen päätoimittajaksi valittiin Tieto. Hankkeen oli tarkoitus päättyä 2013 vuoden lopussa. Tammikuussa 2014 Helsingin Sanomat uutisoi, että järjestelmä myöhästyy ja kustannukset nousevat 500 000 euron verran. [Passi 2014.] Kesäkuussa 2014 Poliisi julkaisi sivustollaan tiedotteen, jossa kerrotaan, että yhteistyösopimus Tiedon kanssa puretaan. Syyksi hankkeen kaatumiselle ilmoitettiin toimitusaikataulungelmat. Tieto maksoi Poliisille 7 500 000 euron kertakorvauksen. Maksu koostui viivästymiskosta, vahingonkorvauksesta sekä nettokauppahinnan palauttamisesta. Poliisi sai myös käyttöönsä tähän asti tehdyn kehitystyön. [Poliisihallitus 2014.]

Tieto Oy:n mukaan ennakoitua suurempi työmäärä aiheutti haasteita. Poliisitarkastajan mukaan haasteet taas johtuivat ”riittämättömästä määrittelystä hankkeen teknisen työn osalta.” Esimerkiksi dokumentaation laatu ei ollut riittävällä tasolla. Tässä vaiheessa Vitja-hankkeesta oli toteutettuna kolmasosa. Hankkeen loppu päätettiin kilpailuttaa uudelleen. [Linnake 2014.]

Vuonna 2015 toukokuussa aloitettiin uuden toimittajan kilpailutus. Uusi hanke päätettiin toteuttaa seitsemässä osakokonaisuudessa, jotka kaikki kilpailutettaisiin erikseen osissa. Hankkeeseen oli tässä vaiheessa käytetty 14 miljoonaa euroa. Ennen uuden kilpailutuksen aloittamista hankkeen johto teki perusteellisen selvityksen vaatimuksista ja konsultoi myös ulkopuolisia tahoja. Vuoden 2015 lokakuussa ilmoitettiin, että Vitjan ensimmäisen osan, eli ytimen ja tiedustelujärjestelmän toimittajaksi valittiin Capgemini. Sopimuksen arvo on noin 2 miljoonaa euroa. [Storås 2015.] Näillä näkymin järjestelmän huomattavin osa, uusi rikosilmoitusjärjestelmä, valmistuu vuoden 2018 lopussa [Vänskä 2015]. Seuraavassa luvussa käsittelen julkisten IT-hankintojen avainongelmia, sekä esitän ratkaisuehdotuksia esimerkkitapauksissa ilmenneisiin ongelmiin.

## **5. Julkisten IT-hankintojen ongelmia ja parannusehdotuksia**

Julkisten IT-hankintojen ongelmat ovat monimutkaisia ja sidoksissa toisiinsa, joten ei ole mahdollista löytää yhtä yksittäistä syytä hankintojen ongelmille. Hankintalakia kritisoidaan usein epäonnistuneiden järjestelmähankintojen yhtey-

dessä. Lainsäädäntöä pidetään syynä siihen, ettei julkisella sektorilla voida hankkia sitä, mitä halutaan tai tarvitaan. Todellisuudessa suurempi ongelma on hankintayksikön valmistautuminen hankintaprosessiin. Mitä isommasta ja monimutkaisemmasta hankinnasta on kyse, sitä paremmin hankintayksikön pitäisi etukäteen selvittää tarpeensa ja suunnitella prosessin kulku. [Eskola *et al.* 2017, s. 24.]

Hankintalaissa on hankintojen suunnittelua koskeva lakipykälä, joka käytännössä tarkoittaa sitä, että hankinnasta tulisi tehdä hankintastrategia. Hankintastrategiaa varten täytyy selvittää muun muassa hankinnan tarve, menettelytavat ja palvelulta vaadittava laatu. Vain osa hankintayksiköistä kuitenkaan tekee hankintastrategiaa tai julkaisee sitä toimittajayrityksille. [Eskola *et al.* 2017, ss. 28-30.]

Seuraavaksi käsittelen yleisimpiä ongelmia hankinnoissa ja parannusehdotuksia näihin. Hankintalaki liittyy moneen ongelmaan läheisesti, joten lain käsittely jatkuu ongelmien yhteydessä vielä tarkemmin.

### **5.1. Liian isoja projekteja**

Hankintalakia on kritisoitu sen kankeudesta. Esimerkiksi Suomen suurimman terveyden IT-uudistuksen, Apotin, hankevetäjän Hannu Välimäen mukaan hankinnan kriteerit kiinnitetään liian aikaisin ja niiden muuttaminen myöhemmin tarkoittaisi kilpailutuksen aloittamista nollasta. [Semkina 2015.] Kun järjestelmään vaaditut toiminnot joudutaan määrittämään liian aikaisin, siitä seuraa helposti tarpeettomankin suuria järjestelmiä. Esimerkiksi Apotin tarjouspyynnössä on 2000 sivua ja järjestelmän toimintavaatimuksia yli 4000 [Mikkola 2015].

Osa tarjouspyynnöistä sisältää ehtoja, joita ei ole mahdollista edes toteuttaa. Varsinkin jos toimintoja on tuhansia, on järjestelmän toimittaminen vaikeaa. Tämän vuoksi todennäköisesti suuremmat projektit epäonnistuvat useammin kuin pienet. Standish Groupin Haze-tutkimuksessa valtion suurista eli yli 5 miljoonan euron projekteista vain 13% onnistui. Vastaavasti taas pienistä eli alle yhden miljoonan euron projekteista onnistui 57% [Standish Group 2015, ss. 1-2].

Tämän lisäksi Standish Group arvioi, että järjestelmien toiminnoista 50% käytetään tuskin koskaan, 30% käytetään harvoin ja vain 20% usein [Standish Group 2010, s. 15]. Tämän perusteella voimme päätellä, että suurin osa järjestelmien ominaisuuksista on turhia. Mitä suurempi projekti, sitä enemmän turhia ominaisuuksia. Turhat ominaisuudet tarkoittavat myös ajan ja rahan tuhlausta. Pahimmillaan monimutkaisuus aiheuttaa myös ongelmia. Kohdassa 4.1. esitetyn VR:n lippujärjestelmän ongelmien yhteydessä Tiedon johtaja Ari Järvelä kuvasi

tietojärjestelmää arkkitehtuuriltaan hyvin monimutkaiseksi järjestelmäkokonaisuudeksi, jonka vuoksi vikojen löytäminen oli vaikeaa [It-viikko 2011]. Herääkin kysymys, tarvitseeko lippujärjestelmän olla monimutkainen arkkitehtuuriltaan?

Toinen hyvä esimerkki liian suuresta hankkeesta on aikaisemmin kohdassa 4.2. esitelty Poliisin Vitja-järjestelmä. Viivästymisten ja ongelmien jälkeen hanke päädyttiin toteuttamaan osissa, kun huomattiin, ettei niin suuren järjestelmän toteuttaminen onnistu kerralla.

Yksi ongelma isoissa projekteissa on se, ettei Suomesta löydy montaa toimittajaa niille. Käytännössä Suomalaisista IT-taloista vain Tieto Oy ja muutama muu isompi toimittaja pystyvät toimittamaan isoja projekteja, koska niillä on tarvittavat resurssit kilpailutuksien voittamiseen ja henkilökuntaa suurten järjestelmien tekemiseen. Pienien IT-talojen resurssit eivät riitä suuriin projekteihin. Tällöin Suomessa jää monien pienempien toimittajien osaaminen kokonaan hyödyntämättä. Usein myös käytetään uudestaan samoja toimittajia, joilla on ollut jo aikaisemmin isoja ongelmia projektien toimittamisessa. Esimerkiksi Tieto oli toiseksi suurin ICT-hankkeiden toimittaja vuonna 2016 Valtion tieto- ja viestintätekniikakeskus Valtorin jälkeen (kuva 2), vaikka sillä on ollut ongelmia toimittaa suurempia järjestelmiä, kuten esimerkiksi Poliisin Vitja-järjestelmää.

Uudistetun hankintalain tavoitteena on tuoda parannusta tilanteeseen. Yksi tavoite uudistuksessa on esimerkiksi parantaa pienten ja keskisuurten yritysten mahdollisuuksia päästä mukaan julkisiin hankintoihin [Eskola *et al.* 2017, s. 32]. Tätä haetaan käytännössä esimerkiksi sillä, että hankintalaki ohjaa hankintayksiköitä jakamaan hankinnat osiin. Tilanteessa, jossa hankintayksikkö ei jaa hankintaa osiin, sen on esitettävä perustelut jakamatta jättämiselle [Julkisten hankintojen neuvontayksikkö 2017d]. Käytännössä laki ei kuitenkaan kiellä jättihankintoja.

On selvää, ettei jättimäisestä järjestelmähankinnasta saa pientä vain ominaisuuksia karsimalla. Esimerkiksi terveydenhuoltoon liittyvät järjestelmät, kuten Apotti, ovat suuria kokonaisuuksia, joissa on valtavasti tarvittavia ja tärkeitä toimintoja. Osissa hankkiminen ei kuitenkaan sovi jokaiseen tapaukseen ja se vaatii tilaajalta pitkäjänteisyyttä, suunnitelmallisuutta ja hankintataitoa. Jokaisen osan kilpailutuksessa erikseen on paljon työtä ja tilaajan on myös huomioitava osien yhteensopivuus keskenään. Patanakulin tutkimuksessa [2014], jossa tutkittiin suuria julkisia IT-projekteja Yhdysvalloissa, Iso-Britanniassa ja Australiassa, kahdeksassa tapauksessa neljässätoista oli ongelmia järjestelmien yhteensopivuudessa. Ongelmana oli esimerkiksi järjestelmän kompleksisuuden ymmärtäminen ja puutteet järjestelmäarkkitehtuurissa. [Patanakul 2014, ss. 21-35.] Tilajan taidot sekä organisaation järjestelmäkokonaisuuden ymmärtäminen ja sen

suunnitelmallisuus ovat siis suuressa roolissa osissa hankkiessa, mutta ylipääntänsä myös jokaisessa järjestelmähankinnassa tulisi huomioida yhteensopivuus.

Osissa hankkiminen tuo siis omat haasteensa, mutta oikein toteutettuna hankinnan jakaminen osiin vähentää suurten projektien epäonnistumisriskiä, sekä mahdollistaa myös pienempien yritysten osaamisen hyödyntämisen. Ketteryys ja osissa hankkiminen on tärkeää myös siksi, että pahimmillaan suuret projektit ovat vanhentuneita valmistuessaan alan nopean kehityksen vuoksi. Tämän vuoksi käsittelen seuraavaksi vanhentuneita projektinhallintamenetelmiä.

## **5.2. Vanhentuneet projektinhallintamenetelmät**

Monia projekteja toteutetaan kankeiden vesiputousmenetelmien avulla, joka esimerkiksi tarkoittaa sitä, että vaaditut toiminnallisuudet tulee tietää ennen hankintaa. Ketterissä menetelmissä voidaan reagoida nopeasti muutoksiin ja määrittää ja muuttaa ominaisuuksia kesken kehitystyön. Ohjelmistosta pyritään tekemään julkaisukelpoisia ja testattavia versioita tiheästi. Valmistuneita osia voidaan testata käyttäjällä ja muuttaa palautteen perusteella, vaikka muu järjestelmä olisi vielä keskeneräinen.

Standish Groupin CHAOS-raportissa ketteriä menetelmiä käyttäneistä kaiken kokoisista projekteista 39% onnistui, kun vesiputousmallia käyttäneistä vain 11% onnistui. Isoissa projekteissa ketterillä menetelmillä tehdyistä onnistui 18% ja vastaavasti vesiputousmallilla tehdyistä vain 3% onnistui. [Standish Group 2016.] Tämän perusteella on selvää, että ketteriä menetelmiä käyttämällä onnistuminen on todennäköisempää.

Vanha hankintalaki ei käytännössä estänyt ketterien menetelmien hyödyntämistä, mutta se oli haastavaa. On hyvä, että uudet päivitetyt JIT 2015 -ehdot tarjoavat valmiit sopimusehdot järjestelmien hankkimiseen ketterin menetelmin.

Uskoisin, että tämä ongelma on tiedostettu hyvin alalla. Esimerkiksi Tietojärjestelmien hankinta Suomessa 2013 -tutkimuksessa tilaajista 45% oli sitä mieltä, että ketteristä menetelmistä voisi olla hyötyä melko tai hyvin paljon järjestelmähankinnassa. Tilaajista vain 13% ei nähnyt, että ketteristä menetelmistä olisi hyötyä. [Celkee Oy *et al.* 2013, ss. 18-19.]

## **5.3. Toimittajalukko ja suljetut järjestelmät**

Toimittajalukko sitoo tilaajan tiettyyn toimittajaan. Hyvä esimerkki toimittajalukosta on kohdassa 4.1. esitelty VR:n tilanne, jossa VR joutuu tilaamaan päivitykset järjestelmäänsä Accenturelta.

Tietojärjestelmien hankinta Suomessa 2013 -tutkimuksessa tilaajista 33% ei varmista koskaan tai vain harvoin, että järjestelmän arkkitehtuuri ja tekninen toteutus sallivat toimittajan vaihtamisen myöhemmin [Celkee Oy *et al.* 2013, s. 16].

Yleensä lukon taustalla on tilaajan puutteelliset taidot hankkia järjestelmiä [Järvenoja *et al.* 2015, s. 38].

Yksi keino välttää toimittajalukkoa on tilata avoimen lähdekoodin järjestelmä. Avoin lähdekoodi tarkoittaa ohjelmistoa, jonka lisenssi sallii vapaan levittämisen, muokkaamisen ja kopioimisen. Avoimen lähdekoodin järjestelmä mahdollistaa useampia eri kehittäjiä, joka taas mahdollistaa paremmin pienten ja keskisuurten yritysten osallistumisen järjestelmien kehittämiseen. JIT 2015 -ehdoissa on tarjolla valmiit avoimen lähdekoodin sopimusehdot tilaajalle. Avoimen lähdekoodin hankinnat sopivat erityisesti sellaisiin hankintoihin, joihin kohdistuu erityisiä julkisuus- tai läpinäkyvyysvaatimuksia. [Järvenoja *et al.* 2015, ss. 36-49.]

Pelkkä avoimen lähdekoodin hankinta ei poista mahdollisuutta lukolle, sillä tämän lisäksi tarvitaan myös suunnitelmallisuutta ja hyvää ICT-johtamista. Näiden rooli on tärkeä, sillä jos organisaation järjestelmien ja hankintojen kokonaisuutta ei ole suunniteltu järkevästi joudutaan lukkoon ajan kuluessa joka tapauksessa. Yksi avoimen lähdekoodin hankinta ei pelasta kokonaisuutta, jos kokonaisuus on muuten suunniteltu heikosti. Tilaajaorganisaatioiden on tärkeää ymmärtää, että ICT-johtaminen on pitkäjänteistä työtä. [Järvenoja *et al.* 2015, ss. 36-49.] Seuraavaksi käsittelenkin tilaajan hankintataitoihin liittyviä puutteita.

#### **5.4. Tilaajan hankintataidoissa puutteita**

Kun tilaajan taidoissa on puutteita, voidaan muun muassa joutua toimittajalukkoon hankinnassa. Tilaaja ei välttämättä ymmärrä ohjelmistomaailman lainalaisuuksia, mikä johtaa siihen, ettei välttämättä täysin tiedetä, mikä on mahdollista toteuttaa ja mikä ei.

Tietojärjestelmien hankinta Suomessa 2013 -tutkimuksessa tilaajat olivat kuitenkin varsin tyytyväisiä omaan hankintataitoonsa, sillä jopa 42% tilaajista piti omaa hankintaosaamistaan hyvänä tai kiitettävänä. Tilaajista 40% piti hankintaosaamistaan tyydyttävänä ja välttävänä 18% vastaajista. Tulokset olivat päinvastaisia, kun kysyttiin toimittajien mielipidettä tilaajien hankintaosaamisesta. Vastaajista kukaan ei pitänyt sitä kiitettävänä, vain 11% piti sitä hyvänä, tyydyttävänä 43% ja välttävänä tai heikkona 46%. [Celkee Oy *et al.* 2013, ss. 12-13.]

Tuloksissa on selkeä ristiriita, sillä tilaajat pitivät omia taitojaan hyvänä, mutta toimittajien mielestä tilaajien taidot olivat enimmäkseen tyydyttävää tai jopa heikkoa. Kallistun tässä enemmän toimittajien puolelle. Tilaajien taitoa ei mairittele esimerkiksi toimittajalukon kohdalla esille tullut tilasto siitä, että 33% tilaajista ei varmista koskaan tai vain harvoin, ettei toimittajalukkoa synny.

Tilaajan tarpeiden selvittäminen ja vaatimusten määrittäminen ovat tärkeä osa hankintaprosessia, sillä jos tilaaja ei tiedä itsekään millaisen järjestelmän hän

tarvitsee, ei toimittajakaan voi sitä tietää. Patanakulin tutkimuksessa [2014] yhdeksätoista tapauksessa neljäsatoista oli ongelmia vaatimusten määrittämisessä. Ne olivat joko puutteellisia tai epäselviä. Syitä puutteellisiin vaatimuksiin oli muun muassa aikapula, käyttäjän näkökulman puuttuminen ja huono dokumentaatio. [Patanakul 2014, ss. 21-35.] Yksi vaihtoehto vaatimusten määrittämisessä onkin kuunnella käyttäjiä ja selvittää sitä kautta järjestelmälle oleelliset prosessit. Järjestelmän hankinnasta vastaava henkilö ei kaikissa tapauksissa itse välttämättä tule käyttämään järjestelmää tai tee työtä samalla tasolla kuin käyttäjät, joten myös vaatimusten määrittäminen voi olla vaikeaa ilman näkemystä.

Ratkaisuna hankintaosaamisen puutteisiin voisi olla konsultaatio riippumattomalta taholta. Huolestuttavaa on kuitenkin, että jopa 74% tilaajista sanoi käyttävänsä harvoin tai ei koskaan riippumatonta asiantuntijaa avuksi kilpailutuksessa. [Celkee Oy *et al.* 2013, s. 17.] Konsultoidessa on kuitenkin hyvä huomioida, että yksityisen sektorin ratkaisut eivät ole välttämättä suoraan sovellettavissa julkisen sektorin tarpeisiin. Heekin [2003] mukaan monet IT-yritykset, konsultit sekä viranomaiset unohtavat tämän hankkiessaan järjestelmiä. Julkisen sektorin tilaajan on siis tärkeää huomioida oman organisaationsa tarpeet ja keskittyä niihin.

## **5.5. Kommunikointiongelmat**

Kun tutkittiin hankkeiden onnistumisen tekijöitä, toimittajat nostivat selvästi tärkeimmäksi tekijäksi tilaajan ja toimittajan välisen viestinnän. Toimittajien mielestä viestintä oli yhtä tärkeää kuin kaksi seuraavaa tekijää yhteensä, jotka olivat toimittajan aktiivisuus ja reagointikyky sekä liiketoimintatavoitteiden toteutuminen. Tilaa- jien listalla viestintä oli toisena, heti hankinnan resursoinnin ja valmistelun jälkeen. [Celkee Oy *et al.* 2013, ss. 5-6.]

Kun tutkittiin kriisiytymisen syitä hankinnoissa, toimittajien mielestä suurin syy kriisiytymisen oli kommunikaation puute. Toisena listalla oli eri näkemys projektin sisällöstä, joka myös liittyy kommunikaatio-ongelmiin osittain. Tilaa- jien mielestä suurimmat syyt kriisiytymiseen olivat aikataulun pettäminen, sekä toisena budjetin pettäminen. Tilaa- jien listalta löytyy myös kommunikaatioon liittyviä ongelmia, kuten eri näkemys projektin sisällöstä ja kommunikaation puute, mutta ei niin korkealta kuin toimittajien listalta. [Celkee Oy *et al.* 2013, ss. 7-8.]

Molempien mielestä kommunikaatio on siis tärkeä osa hankintaprosessia, mutta varsinkin toimittajat kokivat kommunikaation selkeästi tärkeimmäksi onnistumisen tekijäksi ja vastaavasti sen puutteen tärkeimmäksi kriisiytymisen syyksi. Kommunikaatio-ongelmat saattavat johtua edellisessä luvussa esitetystä tilaa- jien taidon puutteesta. Kun ei täysin ymmärretä ohjelmistoalan termejä ja



käytäntöjä, voi olla vaikea keskustella toimittajien kanssa. Sama pätee myös toisinpäin, sillä myös toimittajilla saattaa olla vaikeuksia ymmärtää tilaajan organisaation toimintaympäristön lainalaisuuksia, mikäli ala on vieras. Toinen syy kommunikaatio-ongelmiin on tilaajien ajan ja resurssien puute. Tilaajista 40% ja toimittajista jopa 51% oli sitä mieltä, että tilaajaorganisaation vastuuhenkilölle varatut resurssit olivat harvoin tai ei koskaan riittäviä [Celkee Oy *et al.* 2013, s. 14]. Jos tilaaja ei kerkeä vastaamaan toimittajan kysymyksiin, toimittajat ovat pakotettuja tekemään omia ratkaisuja, jotka eivät välttämättä aja järjestelmää oikeaan suuntaan tilaajan kannalta.

Kommunikaatio-ongelmia on myös havaittavissa alaluvussa 4.2. esitetystä Vitja-järjestelmän hankinnassa. Sopimuksen purkamisen jälkeen Tieto ilmoitti, että ennakoitua suurempi työmäärä aiheutti haasteita. Tämän pohjalta oletettavasti järjestelmän vaatimusten ja toimintojen määrittämisessä on ollut kommunikaatio-ongelmia. Poliisin mielestä haasteet taas johtuivat ”riittämättömästä määrittelystä hankkeen teknisen työn osalta”. Poliisi myös moitti, ettei hankkeen dokumentaation laatu ollut riittävällä tasolla. Molemmilla tahoilla on siis erilainen mielipide kriisiytymisen syyksi. [Linnake 2014.]

Suomessa on myös havaittavissa huonoa kommunikaatiota kuntien välillä. Kunnat hankkivat uudelleen samoja järjestelmiä, jotka on jo toteutettu toisessa kunnassa. Yksi esimerkki tällaisesta tilanteesta on, kun Helsinki ja Vantaa maksoivat kahdella miljoonalla eurolla sähköisen toimeentulotuen asioinnin kehittämisestä, vaikka Kainuussa oli kehitetty vastaava järjestelmä aikaisemmin kahdeksalle kunnalle 400 000 eurolla. Pahimmillaan jokaisessa kunnassa on oma järjestelmänsä ja kun kunnat yhdistyvät, joudutaan hankkimaan kokonaan uudet järjestelmät. Tietyt järjestelmät, kuten esimerkin toimeentulotuen sähköisen asioinnin, voisi toteuttaa kansallisella tasolla eikä jokaisessa kunnassa erikseen. Päällekkäisten järjestelmien hankkiminen on selvää rahan tuhlausta. [Grönholm 2013.]

Kuten edellisessä luvussa selvisi, moni tilaaja ei konsultoi riippumatonta tahoa hankinnoissa. Ulkoisen konsultaation lisäksi järjestelmähankinnoissa kannattaisi enemmän hyödyntää toisten hankintayksiköiden kokemuksia vastaavanlaisista hankinnoista. Esimerkiksi Vitjan ja VR:n hankkeiden toteuttajilla olisi varmasti hyviä neuvoja jaettavana muille järjestelmähankintoja tekeville. Näin kaikkien ei tarvitsisi kävellä samoihin ansoihin.

## **5.6. Väärät tavata mitata onnistumista ja järjestelmien huono käytettävyys**

Tilaajat mittasivat tietojärjestelmähankinnan onnistumista pääasiassa budjetin, liiketoimintatavoitteiden ja aikataulun toteutumisella. Vasta neljänneksi ylsi mitauskohteena loppukäyttäjien tyytyväisyys. [Celkee Oy *et al.* 2013, s. 22.] Voidaan

kuitenkin kyseenalaistaa, onko aikataulun tai budjetin toteutumisella merkitystä, jos valmistunut järjestelmä on laadultaan ja käytettävyydeltään surkealla tasolla.

Esimerkiksi VR:n järjestelmä valmistui noin puoli vuotta myöhässä aikataulusta, mutta pitäisin merkittävämpänä ongelmana pahoja käytettävyyso ongelmia, joita järjestelmästä löytyi. Aluksi järjestelmä ei toiminut ja myöhemmin verkko-kaupasta löytyi yli sata käytettävyyso ngelmaa, kuten kohdassa 4.1. VR:n lippu-järjestelmän yhteydessä kerrottiin [Vanhala 2012; Puskala 2011]. Kohdassa 4.2 esitellyssä Vitja-järjestelmässä myöhästymisen johtui pääasiassa hankintaprosessin virheistä [Linnake 2014]. Suuret projektit, joissa on paljon ominaisuuksia, sisältävät myös paljon enemmän muuttujia, jotka voivat epäonnistua ja johtaa myöhästymiseen. Olisiko järjestelmä myöhästynyt, jos sitä olisi jo alun perin lähdetty toteuttamaan osissa?

Tämä herättääkin kysymyksen, että mikä lopulta on onnistunut ja mikä epäonnistunut järjestelmähankinta? Mielestäni esimerkiksi Standish Groupin käyttämät onnistumisen mittarit, eli aika ja raha, ovat sellaisenaan käytettynä riittämättömiä mittareita mittamaan onnistunutta järjestelmähankintaa. Onnistunut järjestelmähankinta varsinkin julkisella sektorilla on sellainen, joka tekee käyttäjän työstä nopeampaa, helpompaa ja vaivattomampaa. Onnistuneeseen järjestelmähankintaan liittyy myös kokonaisarkkitehtuurin ymmärtäminen. Onnistunut hankinta voisi esimerkiksi olla modulaarinen, helposti tarvittaessa päivitettävä järjestelmä ilman toimittajalukittautumista. Ajallaan ja budjetissa valmistunut, mutta täysin käyttökelvoton järjestelmä ei mielestäni ole onnistunut järjestelmähankinta. Hankinnan tulisi vastata hankintayksikön tarpeisiin, joten laadun ja käytettävyyden tulisi olla hintaa tärkeämpi tekijä hankinnassa.

Heeks [2003] esittää, että hankinnan epäonnistumisen taustalla voi myös olla arvojen erilaisuus. Moni ICT-projekti perustuu koviin arvoihin, kuten esimerkiksi koneisiin, tekniikkaan ja järkevyyteen. Monesti julkiset organisaatiot kuitenkin perustuvat pehmeisiin arvoihin, kuten esimerkiksi ihmisiin, politiikkaan, tunteeseen ja kulttuuriin. Arvojen välille syntyy suuri aukko, kun järjestelmän suunnittelussa käytetyt kovat arvot kohtaavat organisaation pehmeät arvot. [Heeks 2003, s. 5.]

Pahimmassa tapauksessa järjestelmän huonosta laadusta seuraavat asiat ovat vakavia. Esimerkiksi terveydenhoitoalalla järjestelmän laadunpuutokset ja ongelmat voivat vaarantaa potilaan hengen. Valtion tarkastusviraston tutkimuksessa [2011] jopa 43% lääkäreistä oli havainnut työssään tietojärjestelmistä johtuvia haittatapahtumia tai niiden uhan. Kolmannes lääkäreistä koki, että tietojär-

jestelmien virheellinen toiminta oli aiheuttanut tai ollut lähellä aiheuttaa vakavan haittatapahtuman. [Valtiontalouden tarkastusvirasto 2011, s. 52.] Seuraavassa luvussa esittelen ratkaisumallia hankintoihin.

## **6. Ratkaisumalli IT-hankintoihin**

Monella hankintayksiköllä on puutteita hankintastrategioissa. Moe ja muut [2017] tarjoavat ratkaisuksi nelikentän muodossa esitettyä mallia (taulukko 3), jota käytän pohjana omassa ehdotuksessani. Olen sijoittanut nelikenttään esimerkkitapaukset ja myöskin Näkemykseni-kohta sisältää omia huomioitani, miten järjestelmää voitaisiin hankkia kussakin tapauksessa. Nelikentän pohjalta voi lähteä hahmottelemaan hankintastrategiaa.

Nelikentässä järjestelmät on jaettu uniikkeihin ja ei-uniikkeihin järjestelmiin, sekä vaatimuksiltaan monimutkaisiin ja yksinkertaisiin järjestelmiin. Uniikilla järjestelmällä tarkoitetaan järjestelmää, joka on toiminnoiltaan ainutlaatuinen. Yleensä uniikkia järjestelmää käytetään harvinaisella toimialalla. Ei-uniikki järjestelmä on yleisesti käytössä muuallakin. Monimutkaiset vaatimukset sisältävät paljon erilaisia tarpeita ja toimintoja. Tällainen on usein myös käytössä useissa eri organisaatioissa. Yksinkertaiset vaatimukset taas ovat tämän vastakohta eli kun kyseessä on järjestelmä, johon ei tarvita runsaasti toimintoja. [Moe *et al.* 2017, ss. 143-163.]

Hankintastrategiaa suunniteltaessa tilaajan tulisi siis ensiksi selvittää, onko kyseessä uniikki järjestelmä ja kuinka kompleksit vaatimukset ovat. Jos järjestelmä ei ole uniikki, kannattaa perehtyä olemassa oleviin valmisratkaisuihin. Varsinkin mahdolliset avoimen lähdekoodin vaihtoehdot kannattaa huomioida. Esimerkiksi oman palkkajärjestelmän rakentaminen on turhaa, jos on olemassa jo valmiita organisaatioon sopivia tuotteita. Jos kyseessä on ei-uniikki järjestelmä monimutkaisilla vaatimuksilla, kuten esimerkiksi Apotti, modulaarisen ja avoimen järjestelmän hankkiminen on usein paras vaihtoehto. Valmiin järjestelmän hankkimisen pitäisi olla nopeampaa ja halvempaa kuin kokonaan uuden rakentamisen. Tällaisissa hankinnoissa järjestelmän kehittäminen tyhjästä vie aikaa. Tilaajan on kuitenkin tärkeää varmistaa valmistuotteessa, että se soveltuu oman organisaation toimintamalliin ja on yhteensopiva nykyisten järjestelmien kanssa. On myös tärkeää huomioida, että hankitun järjestelmän muokkaaminen ja kehittäminen onnistuvat myös myöhemmin.

Uniikkia järjestelmää hankkiessa on hyvä huomioida, että yksinkertaisista vaatimuksista saa monimutkaiset lisäämällä liikaa ylimääräisiä toimintoja, jotka lopulta paljastuvat turhiksi. Hyvällä hankinnan suunnittelulla, esimerkiksi toimittajalukkoa välttämällä, uusien tarvittavien toimintojen lisääminen on mah-

dollista myös jälkikäteen. Järjestelmän ei usein tarvitse olla monipuolisin heti julkaisussa, kunhan kaikki elintärkeät toiminnot löytyvät. Mielestäni tähän sopii esimerkiksi VR:n järjestelmä, sillä pelkän lippujärjestelmän vaatimuksien ei tarvitsisi olla kovin monimutkaiset.

Taulukko 3. Nelikenttämalli hankintastrategiaan [Moe *et al.* 2017, ss. 143-163].

	<b>Yksinkertaiset vaatimukset</b>	<b>Monimutkaiset vaatimukset</b>
<b>Ei-uniikki järjestelmä</b>	Esimerkki: palkkajärjestelmä.	Esimerkki: Apotti-järjestelmä.
	Suositteltu strategia vaatimusten tarkentamiseen: lainaa tarkat vaatimukset muilta hankintayksiköiltä.	Suositteltu strategia vaatimusten tarkentamiseen: opi muilta hankintayksiköiltä.
	Kommunikaatio toimittajien kanssa: ei välttämätöntä.	Kommunikaatio toimittajien kanssa: käy dialogia toimittajien kanssa järjestelmän määrittämiseksi.
	Näkemykseni: Hanki valmis tuote. Hyödynnä avoimen lähdekoodin vaihtoehdot.	Näkemykseni: Hanki modulaarisen valmis tuote. Hyödynnä avoimen lähdekoodin vaihtoehdot.
<b>Uniikki järjestelmä</b>	Esimerkki: VR:n lippujärjestelmä.	Esimerkki: Poliisin Vitja-järjestelmä.
	Suositteltu strategia vaatimusten tarkentamiseen: Opi muilta hankintayksiköiltä.	Suositteltu strategia vaatimusten tarkentamiseen: Keskustele useamman toimittajan kanssa.
	Kommunikaatio toimittajien kanssa: jatka keskustelua toimittajan kanssa vaatimusten määrittämiseksi.	Kommunikaatio toimittajien kanssa: jatka dialogia toimittajien kanssa, kunnes järjestelmävaatimukset ovat spesifit.
	Näkemykseni: Hanki ketterin menetelmin. Varo toimittajalukkoa. Älä tee projektista liian suurta määrittämällä liikaa ominaisuuksia.	Näkemykseni: Toteuta hanke osissa ketterin menetelmin. Varo toimittajalukkoa. Testaa käytettävyyttä projektin edetessä käyttäjillä.

Uniikki järjestelmä, jolla on monimutkaiset vaatimukset, kuten esimerkiksi Vitja-järjestelmä, kannattaa toteuttaa osissa ketterin menetelmin. Järjestelmän osien käytettävyyttä on hyvä testata käyttäjillä projektin edetessä. Näin saadaan arvokasta palautetta käytettävyydestä ja laadukkaampi järjestelmä.

Vaatimuksien määrittäminen on tärkeää jokaisessa hankinnassa. Vaatimusten määrittämisessä kannattaa hyödyntää myös muiden hankintayksiköiden kokemusta vastaavanlaisista hankinnoista. Erityisen vaativaa vaatimusten määrittäminen on uniikeissa ja vaatimuksiltaan monimutkaisissa hankinnoissa, sillä täysin vastaavanlaisia hankintoja ei ole tehty muissa hankintayksiköissä. Tällaisessa hankinnassa tilaajan kannattaa konsultoida apua ja keskustella useamman toimittajan kanssa, jotta vaatimukset saadaan määritettyä onnistuneesti. Hankkeen toteutuksessa ei kannata kiirehtiä, jos tarpeiden suhteen on vielä epäselvyyttä. Kommunikaation rooli onkin suuri jokaisessa hankinnassa. [Moe *et al.* 2017, ss. 143-163.]

## 7. Yhteenveto

Tutkielmassani olen tutkinut julkisen sektorin IT-hankintoja. Tavoitteenani oli löytää, millaisia ongelmia julkisen sektorin järjestelmähankintoihin liittyy. Esitelen työssäni kaksi julkisen sektorin järjestelmähankintaa, joiden hankintaprosessissa oli ongelmia. Vertailen näissä hankinnoissa ilmenneitä ongelmia tutkimuksiin ja muihin lähteisiin ja sen pohjalta analysoin ongelmia.

Järjestelmähankinta on monimutkainen prosessi, jossa on paljon muuttujia. Näin ollen myöskään ongelmiin ei löydy yhtä yksittäistä vastausta, vaan huomioon on otettava monia asioita. Yksi huomio on se, ettei hankintalaki estä hankkimasta hyvää järjestelmää. Se saattaa aiheuttaa kokemattomalle tilaajalle ongelmia, mutta esimerkiksi JIT 2015 -ehdot tarjoavat valmiita sopimusehtoja hankintoihin. Suuremmat ongelmat löytyvät enemmänkin kommunikaatiosta tai puutteellisista taidoista. Toimittajan ja tilaajan välinen kommunikointi on ensiarvoisen tärkeää järjestelmähankinnassa. Tilaajien olisi myös hyvä huomata mahdolliset puutteet hankintaosaamisessaan ja konsultoida tarvittaessa apua. Järjestelmiä hankkiessa suunnittelu ja pitkäjänteisyys ovat tilaajalle tärkeitä ominaisuuksia, jotta välttään esimerkiksi toimittajalukolta ja yhteensopimattomilta järjestelmiltä. Käyttäjän huomioonottamisella on suuri merkitys, kun tarkoituksena on tehdä käyttäjän työtä nopeuttavia ja helpottavia järjestelmiä. Ketterät menetelmät tarjoavat tähän hyviä työkaluja tiheän testauksen ja tarvittaessa nopeiden muutosten muodossa.

Vaikka työssä käsitellään pääasiassa julkista sektoria, todennäköisesti vastaavanlaisia ongelmia löytyy myös yksityiseltä sektorilta, kuten luvun 3 esimerkeissä huomattiin. Julkisiin hankintoihin käytetään verovarjoja ja järjestelmät tulevat toisinaan myös kansalaisten käyttöön, joten epäonnistumisista uutisoidaan laajemmin. Uskoisin, että osa hankintojen ongelmista tiedostetaan myös hankintaorganisaatioissa, mutta ongelman monimutkaisuuden vuoksi korjaaminen on vaikeaa ja vie aikaa. Tärkeintä olisi muuttaa ajatusmallia hankintojen suhteen.

Tietojärjestelmähankinta ei ole koskaan valmis, vaan se on jatkuva prosessi, jota tarvitsee päivittää ja muokata muuttuvien tarpeiden mukaisesti. Tähän pystyäkseen organisaation tulee ymmärtää omat tarpeensa nyt ja tulevaisuudessa ja rakentaa tämän pohjalta hankintastrategia, joka kattaa nämä tarpeet. Tästä syystä ICT-johtamisen merkitys nousee suureen rooliin organisaatioissa. Yhdellä hyvällä hankinnalla ei ole suurta merkitystä, jos kokonaisuus ei toimi.

## Viiteluettelo

- Jukka Auramies. 2015. *Voi Vitja: Oikeusprosessit jumissa paperipinoissa vielä vuosia – miljoonahanke uusiksi*. MTV. <https://www.mtv.fi/uutiset/kotimaa/artikkeli/voi-vitja-oikeusprosessit-jumissa-paperipinoissa-viela-vuosia-miljoonahanke-uusiksi/5062190#gs.eIC6N2c>. Viitattu 29.11.2017.
- Celkee Oy, Tietotekniikan liitto ry, Ohjelmistoyrittäjät ry. 2013. *Tietojärjestelmien hankinta Suomessa 2013*.
- Saila Eskola, Eeva Kiviniemi, Tarja Krakau ja Erkko Ruohoniemi. 2017. *Julkiset hankinnat*. Alma Talent.
- Pauliina Grönholm. 2013. *Tietojärjestelmissä piilee kustannuspommi*. Helsingin Sanomat. <https://www.hs.fi/kotimaa/art-2000002664959.html>. Viitattu 29.11.2017.
- Richard Heeks. 2003. *Most e-Government-for-Development Projects Fail - How Can Risks be Reduced?*. iGovernment working paper series No. 14. University of Manchester, Institute for Development Policy and Management.
- It-viikko. 2011. *Tietojärjestelmävikä sulki VR:n lippuautomaatit*. Ilta-Sanomat. <https://www.is.fi/digitoday/art-2000001725179.html>. Viitattu 29.11.2017.
- Julkisten hankintojen neuvontayksikkö. 2017a. *Hankintayksiköt*. <http://www.hankinnat.fi/fi/mika-julkinen-hankinta/hankintayksikot>. Viitattu 4.12.2017.
- Julkisten hankintojen neuvontayksikkö. 2017b. *Kynnysarvot*. <http://www.hankinnat.fi/fi/mika-julkinen-hankinta/kynnysarvot>. Viitattu 29.11.2017.
- Julkisten hankintojen neuvontayksikkö. 2017c. *Kokonaistaloudellisesti edullisimman tarjouksen valinta*. <http://www.hankinnat.fi/fi/eu-hankinta/tarjous-ten-valinta/kokonaistaloudellisesti-edullisimman-tarjouksen-valinta>. Viitattu 4.12.2017.
- Julkisten hankintojen neuvontayksikkö. 2017d. *Hankinnan jakaminen osiin*. <http://www.hankinnat.fi/fi/eu-hankinta/hankinnan-kohteen-kuvaus/hankinnan-jakaminen-osien>. Viitattu 4.12.2017.
- Janne Järvenoja, Olavi Köngäs, Karolina Lehto, Jussi Tokola, Marvin von Willebrand ja Kari Wirman. 2015. *JIT 2015 -ehdot: käytännön käsikirja*. Alma Talent.

- Tuomas Linnake. 2014. *Työmäärä yllätti Tiedon, poliisi ei käsitä*. Ilta-Sanomat. <https://www.is.fi/digitoday/art-2000001839331.html>. Viitattu 29.11.2017.
- Ella Mikkola. 2015. *Apotti-hankinnan tarjouspyynnössä tuhansia sivuja ja vaatimuksia*. Yle. <https://yle.fi/uutiset/3-7880025>. Viitattu 29.11.2017.
- Carl Erik Moe, Mike Newman and Mayng Kyaw Sein. 2017. The public procurement of information systems: dialectics in requirements specification. *European Journal of Information Systems* 26, 2, 143-163.
- Minna Passi. 2014. *Poliisin tietojärjestelmäuudistus myöhästyy, luvassa puolen miljoonan euron lisäkustannukset*. Helsingin Sanomat. <https://www.hs.fi/kotimaa/art-2000002705609.html>. Viitattu 29.11.2017.
- Peerasit Patanakul. 2014. Managing large-scale IS/IT projects in the public sector: Problems and causes leading to poor performance. *The Journal of High Technology Management Research* 25, 1, 21-35.
- Poliisihallitus. 2014. *Poliisin tietojärjestelmäuudistus viivästyy - Hallinnon tietotekniikkakeskus (HALTIK) ja Tieto purkavat yhteistyösopimuksensa*. Poliisi. [http://www.poliisi.fi/poliisihallitus/tiedotteet/1/0/poliisin\\_tietojarjestelmauudistus\\_viivastyy\\_-\\_hallinnon\\_tietotekniikkakeskus\\_haltik\\_ja\\_tieto\\_purkavat\\_yhteistyosopimuksensa\\_19458](http://www.poliisi.fi/poliisihallitus/tiedotteet/1/0/poliisin_tietojarjestelmauudistus_viivastyy_-_hallinnon_tietotekniikkakeskus_haltik_ja_tieto_purkavat_yhteistyosopimuksensa_19458). Viitattu 29.11.2017.
- Aapo Puskala. 2011. *VR verkkokaupan käytettävyyssarvio*. <http://www.userpoint.fi/reports/VR%20verkkokauppa%20kaytettavyysarvio.pdf>. Viitattu 29.11.2017.
- Harri Repo. 2011. *VR:n verkkokauppa ei kestänyt lippu-uudistusta*. Tekniikka&Talous. [http://www.tekniikkatalous.fi/talous\\_uutiset/2011-09-14/VRn-verkkokauppa-ei-kest%C3%A4nyt-lippu-uudistusta-3304293.html](http://www.tekniikkatalous.fi/talous_uutiset/2011-09-14/VRn-verkkokauppa-ei-kest%C3%A4nyt-lippu-uudistusta-3304293.html). Viitattu 29.11.2017.
- Juhani Saarinen. 2015a. *VR jäi ohjelmistoyhtiön loukkuun: edessä pakko-ostos vanhalta kumppanilta*. Helsingin Sanomat. <https://www.hs.fi/talous/art-2000002817716.html>. Viitattu 29.11.2017.
- Juhani Saarinen. 2015b. *Uudistus avaa lipunmyynnin yöllä*. Helsingin Sanomat. <https://www.hs.fi/talous/art-2000002817713.html>. Viitattu 29.11.2017.
- Soili Semkina. 2015. *Terveystietojärjestelmän jättimäinen it-hanke saa päätöksen tänään*. Kauppalehti. <https://m.kauppalehti.fi/uutiset/terveydenhuollon-jattimainen-it-hanke-saa-paatoksen-tanaan/wnsRBbAj>. Viitattu 29.11.2017.
- Sofigate, TIVIA, Aalto-yliopisto. 2016. *Tietohallintojen johtaminen Suomessa. Tutkimus 2016*.
- Standish Group. 2016. *CHAOS Report 2015*.
- Standish Group. 2015. *Haze*. [https://www.standishgroup.com/sample\\_research\\_files/Haze4.pdf](https://www.standishgroup.com/sample_research_files/Haze4.pdf). Checked 29.11.2017.

- Standish Group. 2010. *Modernization*. [https://www.standishgroup.com/sample\\_research\\_files/Modernization.pdf](https://www.standishgroup.com/sample_research_files/Modernization.pdf). Checked 29.11.2017.
- Niclas Storås. 2015. *Jättihankkeen rakentaminen alkaa: Poliisi valitsi Capgeminin Vitjan toimittajaksi*. Tivi. [http://www.tivi.fi/Kaikki\\_uutiset/jattihankkeen-rakentaminen-alkaa-poliisi-valitsi-capgeminin-vitjan-toimittajaksi-6001829](http://www.tivi.fi/Kaikki_uutiset/jattihankkeen-rakentaminen-alkaa-poliisi-valitsi-capgeminin-vitjan-toimittajaksi-6001829). Viitattu 29.11.2017.
- Tutkihankintoja.fi. 2017. *ICT-hankinnat*. <https://www.tutkihankintoja.fi/hankintakategoriat/ICT-hankinnat?endDate=31.12.2016&startDate=01.01.2016&lang=fi>. Viitattu 4.12.2017.
- Valtiontalouden tarkastusvirasto. 2011. *Sosiaali- ja terveydenhuollon valtakunnallisten IT-hankkeiden toteuttaminen*. Valtiontalouden tarkastusviraston tuloksellisuustarkastuskertomukset 217/2011.
- Lauri Vanhala. 2012. *Näin VR sotki lippujärjestelmänsä – Miksi it-projektit epäonnistuvat?* Suomen Kuvalehti. <https://suomenkuvalehti.fi/jutut/kotimaa/nain-vr-sotki-lippujarjestelmansa-miksi-it-projektit-epaonnistuvat>. Viitattu 29.11.2017.
- Olli Vänskä. 2015. *14 miljoonaa euroa myöhemmin: "Olemme esimerkki it-hankkeesta niin hyvässä kuin pahassa"*. Tivi. [http://www.tivi.fi/Kaikki\\_uutiset/14-miljoonaa-euroa-myoheemmin-olemme-esimerkki-it-hankkeesta-niin-hyvassa-kuin-pahassa-3221200](http://www.tivi.fi/Kaikki_uutiset/14-miljoonaa-euroa-myoheemmin-olemme-esimerkki-it-hankkeesta-niin-hyvassa-kuin-pahassa-3221200). Viitattu 29.11.2017.



# Pelillisen ja leikillisen oppimisen haasteet ja mahdollisuudet

Arttu Ylhävuori

## Tiivistelmä.

Tutkielma käsittelee kirjallisuuskatsauksen muodossa pelillistä ja leikillistä oppimista esitellen pelillisyyden ja leikkisyyden hyviä ja huonoja puolia koulumaailman kontekstista poimittujen esimerkkien valossa. Lisäksi määritellään pelilliseen ja leikilliseen oppimiseen liittyvät käsitteet tietokoneavusteinen opetus, pelit sekä leikit.

Keskeisimpänä havaintona tutkielmassa havaitaan se, miten pelillinen oppiminen helpottaa arviointikäytäntöjä ja miten sen avulla kodista tulee yksi oppimisympäristö, jossa myös huoltajat voivat seurata lapsensa edistymistä asioiden oppimisessa.

Leikillisessä oppimisessa puolestaan keskeisimpänä havaintona esiin nousee se, miten nuorten ja aikuisten leikkiminen on hyvin vähäistä. Aikuisen leikkimistä aliarvostetaan ja koulutyössä pidemmälle edetessä leikkimistä on yhä vähemmän. Pelinomaiset käytännöt voisivatkin olla ratkaisu ihmisten leikkisyyden puuttumiseen.

**Avainsanat ja -sanonnat:** pelillinen oppiminen, leikillinen oppiminen, tietokoneavusteinen opetus, pelit, leikit.

## 1. Johdanto

Tässä kirjallisuuskatsauksessa vertailen kahta oppimisen käsitettä, jotka ovat *pelillinen oppiminen* ja *leikillinen oppiminen*. Tulevana tietotekniikan ja matematiikan aineenopettajana minua kiinnostaa erityisesti tietokoneavusteisen opetuksen hyödyntäminen – kaikki kyseiseen aihepiiriin liittyvä sisältö ja pelillisyydestä kaikkien mahdollisuuksien hyödyntäminen osana opetusta. Tarkastelenkin aihetta tietokoneavusteisen opetuksen näkökulmasta.

Olen ymmärtänyt, että opetuksen pelillistäminen ja leikillistäminen lisää oppilaiden mielenkiintoa opetettavaan aiheeseen. Epäselvää itselleni on kuitenkin se, miten alan asiantuntijat ovat määritelleet nämä käsitteet ja miten ne eroavat määritelmällisesti toisistaan. Toivon, että tästä kokoavasta selvitystyöstäni on hyötyä itseni lisäksi myös muille aihepiiristä kiinnostuneille opetusalan ammattilaisille.

Tässä katsauksessa tulen määrittelemään tutkielman keskeisimmät käsitteet. Määrittelyn jälkeen vertailen tutkielman keskeisimpien käsitteiden yhtäläisyyksiä ja eroja toisistaan. Tämän jälkeen selvitän käsitteiden käytännön merkityksiä löytämieni reaalimaailman esimerkkien valossa. Lopuksi kokoan

yhteen tutkielman sisältöä, minkä lisäksi esitän pelillisen ja leikillisen oppimisen hyvät ja huonot puolet.

## 2. Keskeisimmät käsitteet

### 2.1. Tietokoneavusteinen opetus

Mitä tietokoneavusteinen opetus sitten on? Tertsusen mukaan *tietokoneavusteisella opetuksella* tarkoitetaan digitaaliseen muotoon tallennettujen oppimateriaalien – siis *tietokoneavusteisten opetusohjelmien* – käyttämistä erilaisissa oppimistilanteissa. Tietokoneavusteinen opetus on yksi opettajien käytettävissä olevista opetusmenetelmistä – oppilaille se on puolestaan työtapa oppia erilaisia ammatillisia tietoja ja taitoja. Opetusmenetelmän ja työtavan näkökulmasta tietokoneavusteisen opetuksen etuna on sen mahdollistama luonteva lähestymistapa käyttää tietoteknisiä laitteistoja ja järjestelmiä oppimisen apuvälineinä. [Tertsunen 1999.]

Lifländerin [1989] määrittely *tietokoneavusteisesta opetuksesta* (engl. muun muassa computer managed learning ja computer based education) on seuraavanlainen. Laajasti määriteltynä tietokoneavusteinen opetus on kaikenlaista opettamista, jossa tietokone on apuvälineenä – opettajan työvälineenä, oppimateriaalin välittäjänä ja oppimisprosessin ohjaajana. Lifländer tekee myös suppeamman määritelmän tietokoneavusteisesta opetuksesta. Kyseessä on siis tietynlainen tietokoneen opetuskäyttö, jossa tietokone toimii oppimateriaalin välittäjänä ja oppimisprosessin ohjaajana – esimerkiksi itseopiskeluun ja itsenäiseen ryhmäopiskeluun soveltuvat opetusohjelmat tai oppimisympäristöt hyödyntävät tietokoneavusteista opetusta. [Lifländer 1989.]

### 2.2. Pelit

Hyvä kysymys voisi myös olla, mitä pelit oikeastaan ovat. Harviainen ja Meriläinen [2013] määrittelevät *pelit* seuraavasti: ne ovat kaikenlaista vapaaehtoista toimintaa, jossa yhdessä sovittujen sääntöjen mukaan tehdään asioita, jotka tuntuvat hankalilta, mutta samalla mielekkäiltä. Lisäksi tässä toiminnassa on yleensä selkeästi havaittava lopputulos.

Mannisen [2011] mukaan peli on järjestelmä, jossa pelaaja sitoutuu vapaaehtoisesti ratkomaan keinotekoista ongelmatilannetta, jonka määrittelevät pelin säännöt – pelin ratkaisusta on puolestaan luvassa mitattavissa oleva lopputulos. Pelissä on usein matemaattinen, logiikkaan perustuva rakenne, joka tavoittelee pelaajan viihdyttämistä. Pelin avulla pelaaja voi kehittyä ja osoittaa taitonsa hallita järjestelmää. Pelin perusrakenteena on siis ohjelmoinnin myötä ohjelmoitu hyvin matemaattinen toiminnallinen runko, jonka ympärille pelin säännöt, ominaisuudet ja looginen rakenne muodostetaan. Pelaajalle on tärkeää, että audiovisuaalinen sisältö on pelissä laadukasta samoin kuin hyvin optimoitu ohjelmistoalgoritmi. [Manninen 2011.]

Tietokonepelit ovat arkipäivää – moni pelaa oppiakseen. Pelit ovat arvokas tutkimusväline ja tutkimuskohde, toimiva opetuksen apukeino sekä lähde opiskelijoiden motivointiin. Sen lisäksi, että pelit ovat nuorison viihteellisen ajankäytön välineitä, nykyään hyödynnetään pelejä ja pelillisiä ominaisuuksia myös muuhun käyttöön. Manninen [2011] ennustaakin, että yhteiskuntamme peruspalvelut ja sovellukset tulevat muuttumaan vähitellen pelinomaisiksi. Jotta pelillisyyden mahdollisuudet selviävät, tulee yrityksen ja erehdyksen avulla tehdä kokeiluja.

Harviainen ja Meriläinen [2013] kuvaavat digitaalisia pelejä seuraavalla 11 kohdan määrittelyllä: Pelit ovat 1) itsenäisiä kulttuuriteoksia, jotka ovat samanarvoisia elokuvien, musiikin ja kirjallisuuden kanssa; 2) suosittu ja hyödyllinen harrastus; 3) tarinoita, joissa peli on tarinan välittäjä; 4) väline omien tarinoiden ja leikkien luomiseen; 5) sosiaalisen vuorovaikutuksen aihe ja alusta; 6) intohimon kohde ja persoonallisuuden määrittelyn väline; 7) teknologista innovaatiota; 8) pelottavien teemojen käsittelyn kanava; 9) perhettä viihdyttävää toimintaa; 10) urheiluväline sekä 11) hyödyllisiä välineitä esimerkiksi oppimiseen ja liikuntaan. Näistä erityisesti 11. kohta on oman mielenkiintoni keskiössä, ja käsittelenkin liikunnallisia sekä pelillisiä ratkaisuja myöhemmin luvussa 4.

Spires [2008] esittelee käsitteen *serious games* (suomeksi vaikkapa ”hyöty-pelit”), joita käytetään yhä enemmän osana nykypäivän opetusta. Kyseisillä peleillä on mahdollisuuksia muuttaa tapaa, jolla katsomme oppimista tämän päivän vaatimuksiltaan jatkuvasti muuttuvassa maailmassa. Serious games -käsitteelle ei ole olemassa yhtä määritelmää – termi kuvastaa yleensä harjoittelua, mainontaa, simulointia sekä koulutusta.

Pelit voivat edesauttaa lasten oppimista ja työskentelyä kouluympäristössä. Pelien jotkin elementit voivat innostaa lapsia opettamaan jotain pelien hahmoille. [Ketamo 2014] Tässä tutkielmassa esitelläänkin kohdassa 4.3 pelihahmon opettamista hyödyntävä SmartKid Maths -oppimispeli.

### 2.3. Leikit

Miten puolestaan leikki liittyy edellä mainittuun peli-teemaan? Mannisen [2011] mukaan ihmiskunnalla on ollut aina tarpeena leikkiä ja pelata. *Leikki* on oleellinen osa ihmisen toimintaa taustasta ja kulttuurista riippumatta. Yksi leikin muoto ovat digitaaliset pelit, jotka antavat haasteita ja onnistumisen kokemuksia. [Manninen 2011.] Toisin sanoen pelit ovat osa leikkiä – digitaalisia videopelejä pelatessa ihminen tulee samalla leikkineeksi pelillisten haasteiden parissa saaden välillä onnistumisen tunteita.

Varhaislapsuudessa uusia kokemuksia kerätään usein leikin kautta. Leikki on lapselle luontainen tapa tutkia sekä oppia uutta ympäristöstä – pelit ja leikit ovatkin usein keskeisiä varhaislapsuuden oppimisen muotoja. Leikki ja pelit

aikaansaavat hyvän olon tunteen, joka motivoi toimintaan. Toiminnan hauskuus, miellyttävyyys sekä kiinnostavuus ovat leikillisen toiminnan ytimenä ja aikaansaajana. [Harju ja Multisilta 2014.]

## 2.4. Pelillinen oppiminen

Vesterinen ja Mylläri [2014] selventävät *pelilähtöisen oppimisen* (engl. game-based learning) ja *pelillistämisen* (engl. gamification) käsitteitä *pelillisyyys*-käsitteen avulla. He ovat itse halunneet keskittyä pelaamisen ideaan liittyviin näkökulmiin, joiden ei tarvitse liittyä suoraan peleihin. He hahmottavat pelillisyyden koulun kontekstin näkökulmasta alueena, johon kuuluu sisältöjä peleistä ja nykypäivän ajankohtaisista pedagogisista kysymyksistä. Pelillisyyttä voidaan tarkastella myös pelillistämisen kautta. Pelillistämisen kytkeytyminen pelillisyyden käsitteeseen onnistuu Vesterisen ja Myllärin [2014] mukaan siten, että pelillisyydessä on erilaisia toimintamuotoja, jotka voivat olla jo alusta asti pelillisiä tai pelillisyyks voi syntyä niissä luonnollisesti.

Pelejä voidaan helposti käyttää opetuksen tukena: Vesterisen ja Myllärin [2014] mukaan pelit, pelikulttuuri ja pedagogiikan kehittämisen pelilliset näkökulmat tuottavatkin mielekästä projektioppimista. Heidän mukaansa tärkeimmät näkökulmat pelillisyyteen ovat 1) dokumentointiin sisältyvä pelillisyyys, 2) vuorovaikutukseen liittyvä pelillisyyys sekä 3) oppimisen ja opetuksen prosesseihin sisältyvä pelillisyyys. Näistä itseäni kiinnostaa etenkin toinen kohta, ja pyrin tässä tutkielmassa tarkastelemaan erityisesti vuorovaikutuksellisuutta osana pelillistä oppimista.

Pelillisyyks tuo mahdollisuuden oppia "luokkahuoneen tuolla puolen" [Vesterinen ja Mylläri 2014]. Vesterinen ja Mylläri [2014] nostavat pelillisyydestä esiin esimerkin: koulut haluavat lisätä oppimiseen samanlaista kokemuksellisuutta ja immersiiivisyyttä ("uppoutuvuutta"), jota ilmenee peleissä. He lähestyvät pelillisyyden ideaa näkökulmasta, jossa ei todellisuudessa puhuta peleistä: he liikkuvat pelien ja pedagogiikan välillä, jossa koulumaailmaan kytkettäessä voidaan puhua pelillisyydestä. He nostavat esimerkkinä pelillisyydestä esiin koulun arjen ja koulujärjestelmän, jossa pelillisinä elementteinä edetään luokka-asteelta toiselle, opitaan samalla uusia tietoja, taitoja ja toimintalogiikkaa menestyksensä kehittämiseen. [Vesterinen ja Mylläri 2014.]

Ihmisillä on peleistä usein mielikuva, että ne opettavat tietoa [Kangas ym. 2014] ja useita taitoja, esimerkiksi motorisia taitoja, kielitaitoa sekä sosiaalisia taitoja [Noppari 2014]. Tästä huolimatta pelien käyttöä oppimisen apuvälineenä on tutkittu vielä varsin vähän [Ke 2009], minkä vuoksi tieteellinen tutkimus pelien hyödyllisyydestä olisi paikallaan oppimispelien käytön yleistymiseksi kouluissa [Kangas ym. 2014].

Manninen [2011] nostaa esiin ajatuksen, jonka mukaan pelit opettavat tehtäviensä haasteellisuudellaan, vaikeudellaan ja lähes mahdottomuudellaan pelaajilleen, miltä yhteiskunnallisesti hyväksytty epäonnistuminen tuntuu.

Pelaaja oppii Mannisen mukaan pelin taitajaksi tekemällä virheitä – ja samalla pelaaja oppii toimimaan elämässään ratkaisulähtöisesti.

Oppimispelien hyödyt ovat Mannisen [2011] mukaan ilmeiset. ”Pelillisuus mahdollistaa luonnolliset, intuitiiviset ja aktivoivat oppimistilanteet. Peli on erinomainen apuväline, joka fokusoi opittavan asian ytimeen. Peleissä on rajallinen ja kontrolloitu maailma, jonka puitteissa ongelmien jäsentäminen helpottuu. Peli on tapa nähdä ja tehdä asioita käytännössä. Se toimii vahvana motivaattorina, havainnollistavana näkökulmien esittäjänä, toimintaan kannustavana kontekstina ja yhteiseen tavoitteeseen osallistavana välineenä.” [Manninen 2011.]

Myös Ketamo [2014] korostaa oppimispelien hyödyllisyyttä. Opettajan ja oppilaan huoltajan näkökulmasta pelit saattavat tarjota hyödyllistä tietoa oppilaan edistymisestä ja oppimisesta. Opettajalta säästyy oppitunnin kallisarvoista aikaa, kun läksyjä ei tarvitse tarkistaa oppilaskohtaisesti luokassa kiertelemällä. Sen sijaan opettaja voi etukäteen katsoa pelin hallinnointikäyttöliittymän avulla, miten luokka on omaksunut asiat ja minkä aihealueiden opettamiseen hänen kannattaa oppitunnin aikana suuntautua. Opettaja saa siis oppilaiden tulokset reaaliajassa ilman tarkastusprosessia. Kallisarvoisen ajan opettaja voi sen sijaan siirtää oppilaiden henkilökohtaiselle ohjaamiselle ja vaikeiden oppiaiheiden opettamiseen. [Ketamo 2014.]

Ketamon [2014] mukaan pedagogisesti suunniteltu peliympäristö pystyy tarjoamaan sekä elämyksellisen väylän oppimiselle että paljon suuremman arvon näyttämällä opettajalle ja huoltajalle oppilaan osaamat ja vielä harjoittelua vaativat oppiaiheet tavalla, joka linkittyy luontevasti oppilaan vapaa-aikaan. Oppilas ei miellä pelillistä oppimisympäristöä oppimisensa mittaamisvälineeksi. Vaikka oppimisen mittaaminen lisää negatiivista kilpailua, aiheuttaa paineita ja saattaa luokitella oppilaita, tarjoaa opettajaa ja huoltajia tukeva analytiikka mahdollisuuksia edistää oppilasta entistä parempiin oppimistuloksiin. [Ketamo 2014.]

## **2.5. Leikillinen oppiminen**

Käsitteenä leikillinen oppiminen voidaan selittää monin eri tavoin. Gordon [2012] on määritellyt *leikillisen oppimisen* (engl. playful learning) leikin aikana tapahtuvaksi oppimiseksi. Hänen mukaansa pelillinen oppiminen erottaa toisistaan vapaan leikkimisen ohjatusta leikkimisestä, joista ohjattu leikkiminen edistää hänestä parhaiten oppimista. Spikol ja Milrad [2008] yhdistävät puolestaan leikilliseen oppimiseen mobiilipeleillä opiskelun ja peleistä oppimisen.

Kankaan [2014] mukaan *leikillisyyttä* (engl. playfulness) on usein verrattu leikkiin, luovuuteen ja emotionaaliin tekijöihin. Kangas on aiemmin määritellyt luovan ja leikillisen oppimisen ”sekä ajatteluun, aktiiviseen tekemiseen että koko kehon hyödyntämiseen perustuvaksi oppimiseksi”. Luovassa, leikillisessä oppimisessä keskiössä ovat luovuus, leikillisuus, tarinallisuus,

yhteisöllisyys, emotionaalisuus, fyysinen aktivisuus sekä monipuolinen teknologian ja median käyttö. [Kangas 2010.]

Harju ja Multisilta [2014] puolestaan ovat muotoilleet leikillisen oppimisen toimenpiteeksi, joka tapahtuu yhteistyössä ympäristön kanssa leikillisistä oppimateriaaleista koostuvassa oppimisympäristössä. Esimeriksi kun oppilaat pelasivat Omaa tietoa etsimässä -lautapeliä (ks. kohta 5.2), he olivat vuorovaikutuksessa keskenään ja käyttivät oppimisympäristönään lautapeliä liittäen siihen leikillisiä elementtejä - kyseessä voisi siis tulkita olevan leikillinen oppimateriaali.

Mannisen [2011] mukaan oppimispelien suunnittelussa pelisuunnittelijaa auttaa oppimisen asiantuntija, jonka tehtävänä on pitää huolta siitä, että peli on toimiva ja siitä saa hyötyä. Peli on eräs leikin muoto, ja onnistuessaan peli sisältää riittävästi leikin piirteitä [Manninen 2011].

Ketamon mukaan leikki, askartelu ja tarinat kuuluvat lapsen maailmaan. Flipped Learning -oppimistapaa ("käänteinen opetusmenetelmä") käyttämällä voidaan hyödyntää tarinoita (kuten videoita, tekstejä ja sarjakuvaa) ennen pelin aloittamista ja askartelua ja leikkiä pelin pelaamisen jälkeen. Kaikkien näiden osa-alueiden tulee herättää mielenkiintoa. Leikkimällä opitut taidot vahvistuvat, ja niistä tulee osa arkipäiväistä toimintaa. [Ketamo 2014.]

### **3. Käsitteiden vertailua**

Mannisen mukaan videopelit jakavat mielipiteitä ihmisten keskuudessa. Lasten ja nuorten pelaamista pidetään useita riskejä sisältävänä, ja peleistä käydään väkivalta- ja riippuvuuskeskusteluja. Pelaamisen vaaroina nähdään sosiaalinen eristäytyminen, elämän hallinnan menettäminen, todellisuudentajun kadottaminen sekä yleinen passiivisuus muilla elämän osa-alueilla. Mannisen mukaan tärkeää on huomioda ikärajat peleissä - kaikkia pelejä ei ole tarkoitettu lapsille. [Manninen 2011.]

Manninen kuvaa osuvasti, miten pelillisyyys on osa leikillisyyttä. "Ihminen on kautta historian kokenut tarpeen leikkiä ja pelata. Leikki on olennainen osa toimintaamme kaikissa kulttuureissa maanosasta ja etnisestä taustasta riippumatta. Digitaaliset pelit edustavat vain yhtä leikin muotoa. Ne tarjoavat haasteita ja onnistumisen kokemuksia meille kaikille." [Manninen 2011.]

Yleisesti pelit tarjoavat tarkoituksenmukaisen ympäristön ongelmalähtöiseen oppimiseen - ne antavat siis mahdollisuuden oppia oppisisältöjä pelin sisältöä hyödyntäen [Vesterinen ja Mylläri 2014]. Ongelmanratkaisun voidaan katsoa pyrkivän kohti tavoitetta, joka ei ole heti saavutettavissa. Pelit tarjoavat tarkoituksenmukaisia runkoja ongelmien tarjoamiseen oppilaille - peli itsessään on suuri ongelma, joka koostuu pienistä kausaalisista ("syysuhteisista") ongelmista. Ongelman muodostavien haasteiden laatu voi vaihdella paljon. Tavallisesti ongelma voi olla mikä tahansa pelaajan etenemistä rajoittava tekijä pelimaailmassa. [Kiili 2005.]

Pelit ovat luonteva oppimisen apuväline lapsille ja nuorille: opetussuunnitelmallisen sisällön lisäksi pelien avulla oppilas pystyy oppimaan oppimisen tapoja ja omaa oppimisen havainnointia ja ymmärrystä. Kyseessä on siis yksi merkittävimmistä asioista kaikkialla tapahtuvan oppimisen teoretisoinnissa. Peleihin liittyvät oppimisen tavat ovat monille oppilaille luonnollisia vapaa-ajan pelaamisen takia. Pelit ovat hyvä yhdiste oppilaan maailman ja koulun välillä silloin, kun oppilaan omat oppimisen tavat huomioidaan koulussa. [Vesterinen ja Mylläri 2014.]

### **3.1. Pelillisen oppimisen haasteet**

Mannisen [2011] mukaan kun mietitään peliä hyötykäytön kannalta, on oltava tietoinen pelin oleellisimmista laatu-seikoista. Pelin tulee olla haastava, elämyksellinen, mukaansa tempaava sekä viihdyttävä. Vaarana on se, että pelin sisältö on piilotettu ulkoasultaan epäonnistuneeseen, peliksi kutsuttuun sovellukseen. Huono pelillisuus johtaa sisällön jäämiseen epäselväksi. Vaarana voi myös olla se, että pelinä hyvin toimivat sovellukset ovat monesti sisällöllisesti liian suppeita – peli voi olla irti todellisuudesta ja arjesta. [Manninen 2011.]

Oppimispelien käyttöönoton ja toimintatapojen omaksumisen haasteina on Mannisen [2011] mukaan ajan, energian ja radikaalien muutosten puute pedagogisessa ajattelussa. Haasteena on myös oppimispelien rakentamisen vaatima vahva pohja pedagogisessa osaamisessa ja pelisuunnittelussa sekä riittävät rahalliset investoinnit pelien kehitykseen. Tähän ratkaisuna voisi olla esimerkiksi se, että oppilaat suunnittelevat itse omia oppimispelejä oppimateriaaleikseen.

Mannisen tavoin myös Vesterisen ja Myllärin [2014] mukaan oppimispelien hyödyntämisen pitäisi olla mielekästä pedagogisessa mielessä. Tarkastelun pitäisi kattaa paremmin pelit osana pedagogisia kokonaisuuksia ja muuta oppimisympäristöä. Haasteena on useinkin se, miten pelit saadaan kytkettyä opetuskokonaisuuksiin ilman, että ne jäävät irrallisiksi sekä opetuksen että oppimiskokemuksen näkökulmasta. [Vesterinen ja Mylläri 2014.] Toisin sanoen oppimispeleistä tulisi saada oppimisen kannalta konkreettista hyötyä, jotta niitä olisi järkevää käyttää osana opetusta. Jos opettaja esimerkiksi kokee, että oppimispeli vain riehaannuttaa oppilaat tarjoamatta opetuksellista vastinetta, pelin käyttö saattaa jäädä yhteen kokeilukertaan.

Pelien käyttö osana perinteistä luokkahuonetyöskentelyä saattaa usein sotea luokkahuonetyöskentelyä. Pelit nähdään usein ylimääräisinä viihteellisinä kevennyksinä kaiken muun opiskelun ollessa valmiina. Tällä tavalla peleistä ei saada irti pelien suunnitelmallisen käytön mahdollistamaa hyötyä. [Ketamo 2014.]

### **3.2. Leikillisen oppimisen haasteet**

Manninen [2011] on havainnut, että aikuisen leikkimistä usein aliarvostetaan – leikkiä ei osata nähdä oleellisena osana ihmiskulttuuria toisin kuin aikuismaista

ja ammattimaista toimintaa. Leikki on oleellinen osa alaluokkien opetusta, mutta Mannisen tekemien havaintojen mukaan koulutyössä pidemmälle edetessä leikkimistä on yhä vähemmän. Esikoulutasolta alkaen ylempiä luokka-asteita tarkastellessa hän on tehnyt seuraavanlaisen huomion: mitä korkeammasta luokka-asteesta on kyse, sitä kaavamaisempaa ja tylsempää luokkahuoneen opetus on. [Manninen 2011.] Toisin sanoen alemmilla koululuokilla leikitään osana opetusta, ylemmillä taas ei enää juurikaan.

Mannisen [2011] mukaan suurimmat toiminnan rajoitukset tulevat ihmisestä itsestään. Kun luovuus, rohkeus ja innovatiivisuus katoavat, on myös leikillisuus kadoksissa – ei uskalleta olla leikillisiä eikä ajatella leikinomaisesti edessä olevia haasteita. Tähän liittyen Manninen esittelee käsitteen *itseään ruokkivan valheellisuuden kiertokulku* (engl. self-feeding iterative fallacy). Onkin ehdotettu, että oppiminen tulisi aloittaa poistamalla turha valheellisuus, antaa ihmisen olla sitä mitä hän on ja pyrkiä valjastamaan oppijan energia ja tuottavuus uudella tavalla. [Manninen 2011.]

Manninen [2011] esittää, että pelinomaisilla ratkaisuilla ihminen saadaan työskentelemään yhdessä muiden kanssa hyödyntäen leikillisiä keinoja. Inhimillisyys, leikillisuus, unelmahakuisuus, epäonnistumisen kestäminen sekä epävarmuuden sietokyky ovat keskeinen osa pelejä ja pelaamista. Jokaisella ihmisen toiminnan alueella on mahdollista löytää leikillisyyttä. Oppiminen on näistä toiminnan alueista selkein ja Mannisen mukaan yksi tärkeimmistä kehittämiskohteista. [Manninen 2011.]

## **4. Esimerkkejä pelillisestä oppimisesta**

### **4.1. Liikunnalliset pelit**

Kiili ym. [2014] ovat tutkineet *liikunnallisia oppimispeliratkaisuja* (engl. educational exergames), joissa heidän mukaansa yhdistyy fyysinen liikkuminen ja oppimista tukeva kognitiivinen toiminta. Kiilin ja kumppaneiden tutkimus osoitti, että oppilaiden matkapuhelimissa toimivat, pilvipalveluihin perustuvat liikunnalliset oppimispeliratkaisut sopivat koulun toimintaan hyvin.

Aiemmassa tutkimuksessaan Kiili kollegoineen tutki intensiivistä liikuntaa vaativia pelejä käyttäen niitä apuna opittujen asioiden kertaamisessa. Koska intensiivinen liikunta vaikeuttaa kognitiivisia prosesseja, intensiivistä liikuntaa vaativat pelit sopivat parhaiten opittujen asioiden kertaamiseen [Kiili ym. 2012].

Intensiivisenä liikuntapelinä toimi Kiilin ja kumppaneiden [2012] tutkimuksessa peruskoulun alakoululaisille suunnattu Brain Dive –matematiikkapeli, jota pelataan juoksemalla paikallaan. Kyseisestä pelistä saadut käyttökokeemukset ovat olleet positiivisia: tutkittavat oppilaat pelasivat peliä innolla ja matematiikkaa ja liikuntaa yhdistävää tehtävien harjoittelua pidettiin mielekkäänä. Tutkimus näytti toteen sen, että peli kykenee edesauttamaan matema-



tiikan oppimista ja on hyvä vaihtoehto perinteiselle paperiselle tehtäväkirjalle toistoja vaativien tehtävien laskemisessa. [Kiili ym. 2012.]

Kyseisessä liikuntapelitutkimuksessa havaittiin myös ongelmia oppimispelien suunnittelussa. Pelin vaikeimmissa kentissä ja huonokuntoisimmilla pelaajilla kauan kestävässä pelivuoroissa pelaajan kognitiivinen ja fyysinen kuorma nousivat yhtä aikaa liian korkeaksi, jolloin pelaaja ei kyennyt enää yhtä tehokkaasti ratkaisemaan tehtäviä. Tutkijat huomasivat, että yksinkertaisissakin peleissä pelin tarkoituksena olisi ottaa huomioon pelaajan taitotaso sekä fyysinen kunto. Näin pelin mielekkyys säilyy tarkoituksenmukaisena. [Kiili ym. 2014.]

#### **4.2. Pelillisuus koulussa**

Vesterinen ja Mylläri ovat saaneet seuraamassaan Kauniaisten Mäntymäen koulussa huomata, miten ahkera tieto- ja viestintätekniikan pedagoginen käyttö on nostanut pelillisyyden oleelliseksi tekijäksi, joka tukee oppilasprojekteja ja oppimisprosesseja. Kauniaisten Mäntymäen koulussa pelillisuus on otettu seuraavasti huomioon: 1) opetuksessa käytetään pelillisiä materiaaleja ja työvälineitä, 2) opetuksen suunnittelussa käytetään pelillisiä elementtejä ja projektoiminnat pyritään pitämään pelillisinä sekä 3) oppimisprojekteja ohjattaessa sallitaan ja tuetaan lasten omat pelilliset ideat ja pelilliset toimintatavat. [Vesterinen ja Mylläri 2014.]

Pelillisten materiaalien ja työvälineiden käyttö keskittyy Vesterisen ja Myllärin mukaan Mäntymäen koulussa oppimisen osioihin, jotka hyödyttävät oppilaan omaa motivaatiota sekä itseohjautuvuutta – kyseessä on siis oppilaiden yksitoikkoisiksi tai stressiä aiheuttaviksi mieltämien toimintojen muuttaminen mielekkäämpään suuntaan. Oppilas hyödyntää tarinallisia, interaktiivisia sekä etenemistä ja osasuorituksen laatua osoittavia toimintoja ja ylläpitää, valvoo ja ohjaa omaa oppimistaan. Kun pelaaminen on oppilaan omaa toimintaa, opettaja voi huoletta keskittyä eriyttämiseen sekä arvioinnin suunnitteluun. Kun pelillisyyttä hyödynnetään opetuksessa osana tieto- ja viestintätekniikkaa, on kodin ja koulun välinen yhteistyö oleellista eriyttämisen ja arvioinnin kannalta. Oppilaan oppimisprosessi on myös kotiväen nähtävissä. Oppilaan oppiminen ja koulutyöt näkyvät paremmin koulun ulkopuolella – oppilaalla on valta päättää, milloin hän pelaa oppimispeliä ja hän voi helposti jakaa kokemuksiaan pelaamisen etenemisestä huoltajiensa kanssa. [Vesterinen ja Mylläri 2014.]

Mäntymäen koulussa näkyy puolestaan pelillisten elementtien käyttö sekä pelillisten projektitoimintojen käyttö esimerkiksi kuudennen luokan Aasia-projektissa, jossa oppilaat pelasivat verkossa pareittain roolipeliä verkko-opettajina. Oppilaiden tehtävänä oli tutustua yhteen Aasian valtioista, esitellä valtio muulle luokalle verkkokartalta näyttäen (jaettuna oppimateriaalina täydentä-mässä taitotestinä muuta oppimateriaalia) ja toteuttaa verkkolomakepohjainen

tietovisa tästä valtiosta. Oppilaille oli annettu myös tehtäväksi antaa palautetta koko luokan ja yksittäisen oppilaan suorituksesta verkkoalustaa käyttäen. [Vesterinen ja Mylläri 2014.]

Pelillisten ja projektimuotoisten opetustilanteiden etuina ovat Vesterisen ja Myllärin mukaan vertaisvuorovaikutus ja -palaute sekä leikillinen kilpaileminen. Aasia-projektissa motivaatiota tukivat mahdollisuus 1) upottaa oma projekti verkkokarttamerkkiin sekä nähdä muiden työparien projektit, 2) tutkia muiden työparien projekteja ja niiden teknisiä ja kerronnallisia ratkaisuja ja taitotestituloksia sekä 3) antaa ja saada vertaispalautetta. [Vesterinen ja Mylläri 2014.]

Oppilaiden omien pelillisten ideoiden ja pelillisten toimintatapojen salliminen ja tukeminen näkyi Mäntymäen koulun opetuskokeilussa, jossa käytettiin pilviteknologiaa, mobiililaitteita sekä Citynomadi-matkailusovellusta (jota oppilaat käyttivät paikallishistorian oppimiseen älypuhelimien sijaintitiedon avulla). Projektissa pelillisyyden luontainen syntyminen prosessin aikana on oppilailla oman pelillisen kerroksen luomista. Siinä oleellista olivat työskentelyn etenemisen vertaaminen kilpailullisesti sekä media- ja pelikulttuurista tuttujen kieli- ja vertauskuvien käyttö. [Vesterinen ja Mylläri 2014.]

Oppilaiden tehtäväksi oli annettu Kauniaisissa kulkeminen ja lähiseudun historiaan liittyvien tehtävien suunnittelu luokkakavereille. Kun oppilaat alkoivat taktikoida ja pyrkiä hämäämään muita ryhmiä käyttäessään Citynomadi-sovellusta ja GPS-merkintäteknikkaa, he vertasivat ryhmänsä toimintaa The Amazing Race -tv-sarjan kilpailuun. Oppilaat loivat siis pelillistä tilannetta kommunikaationsa tuloksena.

Opettajat puolestaan tukivat projektin aikana syntynyttä pelillistä toimintaa suunnittelemalla tehtäviin liittyvän pistelaskun sekä finaalilähetysten, jossa oppilaat antoivat motivaatiota nostattavaa pelityylistä vertaispalautetta. Finaalilähetyksessä käytettiin pelillisiä termejä ”kilpailijoiden kuuleminen” tuomarineuvoston toiminnan yhteydessä sekä ”tuomion” antaminen ryhmän suorituksen pohjalta. [Vesterinen ja Mylläri 2014.]

#### **4.3. SmartKid Maths -peli**

SmartKid Maths -oppimispelin ideana on se, että lapsi opettaa pelihahmolleen matematiikkaa hankkimalla opetettavaa tietoa. Väärin opettaminen ei ole virtuaalisen pelihahmon tapauksessa vahingollista toisin kuin toiselle lapselle vahingossakin väärin opettaminen, joka saattaa tuottaa molemmille lapsille väärinkäsityksiä. Näiden virheellisten käsitysten oikeaksi korjaamisessa saattaa kestää kauan. Pelihahmojen oppimiseksi lapset ovat valmiita tekemään paljon työtä – siis paljon harjoituskertoja, mikä lisää Ketamon [2014] mukaan oppimisen syvyyttä.

SmartKid Maths -oppimispelin taustalla on pelitekoäly, joka oppii konstruktivista oppimiskäsitystä vastaavalla tavalla pelaajan opettamia käsite-

teita. Konstruktiivisella oppimiskäsityksellä tarkoitetaan tässä yhteydessä maailmankuvan jatkuvaa kehittämistä vuorovaikutuksessa ympäristön ja muiden kanssa. Pelitekoälyn tehtävänä on saada tietokoneen ohjaamat hahmot sekä pelimaailma näyttämään uskottavan oloisilta. Pelitekoälyn näkökulmasta käsitteellinen mallinnus tarjoaa mahdollisuuksia, joita ei muilla menetelmillä ole mahdollista saavuttaa. Pelaajan käsiterakenne voidaan mallintaa pelihahmon kautta käsitekartaksi, jonka ansiosta pelaajan osaamista pystytään simuloimaan pelissä pelin osana tai oppimisen analytiikkaa tulkitsemalla. [Ketamo 2014.]

SmartKid Maths -pelin neljä tärkeintä elementtiä ovat 1) oma pelihahmo, 2) kenttävalikko, 3) valittu opetustila sekä 4) kilpailulabyrintti. Käsittelen näiden ympärille liittyviä seikkoja seuraavissa kappaleissa.

Ensimmäinen oleellinen osa SmartKid Maths -peliä on opettaa omaa pelihahmoa, jonka valitseminen onkin pelin ensimmäinen tehtävä. Valittavana on monia erilaisia hahmoja – oppilas tuntee opettamisen itselleen mieleisemmäksi, kun kyseessä on itse valittu, oma hahmo. Kaikki pelihahmot ovat erialaisia – niillä on oma persoonallisuus ja omanlaisensa visuaalinen ulkonäkö. [Ketamo 2014.]

Toinen keskeinen seikka pelissä on kenttävalikko, josta näkyy pelaajan edistyminen ja suoritustaso. Pelaamista on rajoitettu niin, että alussa pelikenttiä on valittavissa vain muutamia ja uusia aukeaa pelissä edistymisellä. Uusien pelijaksojen ansaitseminen on vapaa-ajalla pelaaville oppilaille ennestään tuttua, mikä nostaa motivaatiota. Ketamon mukaan oppimispelit tuleekin tehdä samalla tavalla pelikerronnallisia tapoja hyödyntäen, vaikka nämä tavat saattavatkin poiketa suuresti opetusalan käsityksistä oppimateriaalien kerronnassa. [Ketamo 2014.]

Kolmas tärkeä osa-alue SmartKid Maths -pelissä on valittu opetustila. Oppilas voi siirtyä kenttävalikosta valitsemaansa opetustilaan, joka sijoittuu yksilöllisesti johonkin maantieteelliseen paikkaan. Kun paikkoja on liitetty peliin, oppilas on kiinnostuneempi ympäristöstään ja oppii ohimennen maapallon kuuluisimpia paikkoja. [Ketamo 2014.]

Opetustilan keskeisenä tehtävänä on myös antaa oppilaalle mahdollisuus opettaa omalle pelihahmolleen tiettyä aihepiiriä noudattavien ongelmien ratkaisemista. Oppilaan tulee auttaa hahmoaan oppimaan, mikä on tälle esitettyjen kysymysten oikeat vastaukset. Oppilas saa heti vastauksensa annettuaan palautteen peliltä, mikä edistää lapsen oppimisprosessia. Välittömästi annetun palautteen tehtävänä onkin auttaa vahvistamaan tai kyseenalaistamaan oppilaan ajatusmallia – samanlaista hyötyä ei luultavasti saataisi myöhemmin annetussa palautteessa. [Ketamo 2014.]

Neljäs oleellinen osa-alue peliä on kilpailulabyrintti, johon oppilas lähettää pelihahmonsa riittävän opettamisen jälkeen. Tällöin hahmo on pelkästään oppilaan antaman opetuksen varassa. Labyrintissä pelihahmo kilpailee virtuaalikel-

pailijoiden kanssa tekemällä pelikentän opetusteeman mukaisia tehtäviä. Jos oppilas on opettanut pelihahmoaan riittävästi, hahmo voittaa kilpailun ja pelin seuraava opetustila aukeaa oppilaalle. [Ketamo 2014.]

Pelin labyrinthi on siitä kätevä, että se visualisoi oppilaan ja pelihahmon opetussuorituksiin liittyviä seikkoja. Pelihahmo juoksee labyrinthissa sitä nopeammin, mitä enemmän käsitteitä ja relaatioita (eli käsitteiden välisiä suhteita) oppilas on hahmolleen opettanut. Samoin pelihahmo ratkoo labyrinthin tehtäviä sitä nopeammin, mitä nopeammin oppilas on opettanut pelihahmoaan. Lisäksi pelihahmo suoriutuu tehtävistä sitä varmemmin, mitä paremmin oppilas on opettanut pelihahmoaan. [Ketamo 2014.]

SmartKid Maths -pelissä eteneminen on suunniteltu siten, että haastavampaan aihepiiriin pääseminen vaatii selviytymisen kannalta riittäviä pohjatietoja – siis aikaisemman tiedon päälle rakentuva uusi osaaminen tehdään näkyväksi ja oppilas ei tunne turhautumisen tunnetta. Taitotason riittäessä peli päästää nopeammin oikean haastavuustason tehtäviin. Palkinnot ja avautuneet uudet kentät näyttävät lapselle etenemisen ja suoritustason pelimäisen kerronnan avulla. Lapsi siis ymmärtää etenemisen ja osaamisen, muttei koe olevansa tarkkailtavana – kyseessä on peleille tyypillinen keino esittää etenemistä ja osaamista. Samanaikaisesti peli tarjoaa huoltajille yhdellä yhteenvedonomaisella kenttävalikon tarkastelulla tiedon siitä, miten oppilas on edistynyt ja omaksunut opiskelemansa aihepiirit. [Ketamo 2014.]

## **5. Esimerkkejä leikillisestä oppimisesta**

### **5.1. Omaa tietoa etsimässä -lautapeli**

Kangas [2014] kuvaa yhteisöllistä peliä käyttäen esimerkkinä neljän kahdeksaluokkalaisen pelitilannetta Omaa tietoa etsimässä -lautapelissä. *Omaa tietä etsimässä* on ensisijaisesti oppilaanohjaukseen tarkoitettu oppimispeli. Se on kehitetty yläkoululaisille antamaan tietoa Suomen koulujärjestelmästä ja estämään koulupudokkuutta ja syrjäytymistä. Pelin on kehittänyt Linnanen yhteistyössä Helsingin Normaalilyseon, Kehittämiskeskus Opinkirjon, Eduplus Oy:n ja Koulu kaikkialla -hankkeen kanssa. [Linnanen 2014.]

Omaa tietoa etsimässä -lautapelin ideana on tarjota konkreettinen, osallistava ja hauska keino opettaa Suomen koulutusjärjestelmän keskeinen perusrakenne peruskoulusta toisen ja kolmannen asteen opintoihin. Oppimispelin tarkoituksena on tarjota kestävä esittely ja mahdollisuus muodostaa yhdessä ymmärrys maamme koulutusjärjestelmän rakenteesta. Fyysiseltä rakenteeltaan peli koostuu pelilaudasta, elämäkorteista ja pelaajalautoista. Pelaajien tehtävänä on valita vuorollaan kortti, jonka ohjeiden mukaan tulee toimia – esimerkiksi koulujärjestelmässä liikkuminen ja risteyskohdissa valintojen tekeminen. Elämäkorttien tehtävänä on myös määrittää 1) elämänlaatu, 2) varallisuus sekä 3) motivaatio. Kaikkia kolmea arvoa tarvitaan pelissä etenemiseen, ja niistä

pidetään kirjaa pelaajien omilla pelilaudoilla. [Linnanen 2014.] Korttien vaikutuksena pelin kulku saattaa helpottua tai vaikeutua. Esimerkin peliä pelatessa oppilaiden välinen leikillisuus näkyy myönteisenä kommunikaationa, huumorin käyttönä sekä ilmaisullisella sanavalinnoilla leikittelyllä [Kangas 2014].

Esimerkkipeli kuvastaa Kankaan [2014] mukaan leikillistä asennoitumista ja leikillisen tilan rakentumista. Kun oppilas näyttää olevansa asenteeltaan leikillinen, hän kutsuu epäsuorasti toiset mukaan leikkiin. Oppilas siis hyödyn-tää kaiki sanallisen viestimisensä lisäksi nonverbaalisia viestintäkeinoja näyttäessään leikittelevänsä jollakin aihepiirillä. Kankaan mukaan leikillisuus näyttää sallivan epäonnistumisen ja virheiden tekemisen, mikä on hänen mukaansa oleellinen seikka yhteistyössä toimimisen ja oppimisen kannalta [Kangas 2014].

## **5.2. Angry Birds Playground -oppimateriaalit**

Harjun ja Multisillan tutkimuksessa leikillinen oppimisympäristö koostui Rovi-on Angry Birds Playground -oppimateriaaleista. Niiden ”leikillisten element-tien” (joita ovat pelit sekä tutut hahmot) ja ”laadukkaan sekä mielekkään sisällön” avulla lapsia aktivoitiin opettelemaan erilaisia taitoja sekä tutkimaan materiaaleihin liittyviä opittavia asioita, jotka perustuvat soveltaen suoma-laiseen esiopetussuunnitelmaan. Tutkimuksessa käytetyt oppimateriaalit olivat käytännössä puuhakirjoja, piirustusjulisteita, leikkikortteja sisä- ja ulkoleik-keihin, kanteleita, tablet-laitteiden e-puuhakirjoja sekä kännykän yhteenlasku-ja muistipelejä. Näistä osaa käytettiin saman aiheen oppimiseen (esimerkiksi matematiikkaa puuhakirjojen, pelien ja leikkien avulla) ja osan avulla harjoi-teltiin tietyn oppisisällön tarpeita (kuten musiikkia kanteleita soittamalla). Tutkijoiden mukaan monipuolisella oppimateriaalien hyödyntämisellä jokai-nen lapsi voi löytää itselleen mieluisen tavan tutkia ja oppia uutta – tietyn aiheen monipuolinen opettelu materiaalien ja aktiviteettien avulla saattaa vahvistaa vahvojen käsite rakenteiden syntymistä sekä edistää lapsen oppimis-ta. [Harju ja Multisilta 2014.]

## **6. Yhteenveto**

Tässä tutkielmassa olen käsitellyt pelillistä ja leikillistä oppimista keskittyen pääasiassa alakoulumaailmaan. Olen saanut huomata, että leikilliseen oppimi-seen liittyviä tieteellisiä artikkeleita vaikuttaisi olevan reilusti vähemmän kuin pelillistä oppimista käsitteleviä. Lieneekö syynä se, että oppimisen kontekstissa pelillisuus lienee trendikkäämpi aihe kuin leikillisuus vai vain oman kiinnos-tukseni painottuminen pelillisyyttä käsitteleviin lähdemateriaaleihin.

Tässä tutkielmassa on määritelty oleelliset käsitteet määritelmien vi-vahde-erot huomioiden. Tähän yhteenvetoon kiteytän vielä lyhyesti, mitä mikäkin käsite tarkoittaa.

Tietokoneavusteinen opetus on digitaalisten oppimateriaalien käyttämistä oppimistilanteissa. Ne ovat luonteva lähestymistapa tietoteknisten laitteistojen ja järjestelmien käyttämiseen oppimisen apuvälineenä.

Yhden määritelmän mukaan pelit puolestaan ovat järjestelmä, jossa pelaaja sitoutuu vapaaehtoisesti ratkomaan keinotekoisia ongelmatilannetta, jonka määrittelevät pelin säännöt. Leikki taas on etenkin pienelle lapselle luontainen tapa tutkia ja oppia uutta ympäristöstä.

Pelillinen oppiminen (tai lähdemateriaalin käyttämä termi *pelillisuus*) osana koulumaailmaa tarkoittaa aluetta, johon kuuluu sisältöjä peleistä ja nykypäivän ajankohtaisista pedagogisista kysymyksistä. Leikillinen oppiminen taas voidaan yhden määritelmän mukaan leikin aikana tapahtuvaa oppimista, joista ohjattu leikkiminen saattaisi edistää parhaiten oppimista.

Esimerkkinä mainitsin muun muassa liikunnallisen Brain Dive -matematiikkapelin. Sen avulla tehty tutkimus osoitti, että peli kykenee edesauttamaan matematiikan oppimista, ja se on hyvä vaihtoehto perinteiselle paperiselle tehtäväkirjalle toistoja vaativien tehtävien laskemisessa. Tosin huonona puolena tutkimuksessa havaittiin, että vaikeimmissa kentissä ja huonokuntoisimmilla pelaajilla kauan kestävässä pelivuorossa oppilas ei kyennyt yhtä hyvin ratkaisemaan tehtäviä – huonokuntoiset ja heikosti matematiikkaa hallitsevat pelaajat eivät näet pärjää pelissä. Tutkijat totesivatkin, että yksinkertaisissa peleissä pelin tarkoituksena olisi ottaa huomioon pelaajan taitotaso sekä fyysinen kunto pelin mielekkyyden säilyttämiseksi.

Pelillisen oppimisen selkeimpänä hyvänä puolena on se, miten arviointikäytännöt helpottuvat ja kodista tulee yksi oppimisympäristö, jossa myös huoltajat voivat seurata lapsensa edistymistä asioiden oppimisessa. Oppimispelit tarjoavatkin hyödyllistä tietoa oppilaan edistymisestä ja oppimisesta.

Toinen selkeä etu pelillisyydessä on hyväksytty epäonnistuminen osana oppimisprosessia. Pelaamisessa virheiden tekeminen on täysin sallittua, joita tekemällä pelaaja oppii sekä pelin taitajaksi että toimimaan elämässään ratkaisulähtöisesti.

Pelillisen oppimisen selkeänä huonona puolena on ongelmallisuus pelin laatuskoissa ja pedagogisessa mielekkyydessä. Pelin sisältö saattaa olla piilotettu ulkoasultaan epäonnistuneeseen, peliksi kutsuttuun sovellukseen, jonka sisältö on epäselvä. Lisäksi vaarana on se, että pelit jäävät irrallisiksi sekä opetuksen että oppimiskokemuksen näkökulmasta eikä niitä saada kytkettyä opetuskokonaisuuksiin. Oppimispelien tulisikin olla samankaltaisia kuin "oikeidenkin" viihteellisten pelien, vaikka niiden tavat saattavatkin poiketa suuresti opetusalan käsityksistä oppimateriaalien kerronnassa.

Leikillisen oppimisen hyvänä ja hieman yllättävä havaintona puolestaan on se, miten peli ja leikki linkittyvät toisiinsa. Peli on eräs leikin muoto, ja onnistuessaan peli sisältää riittävästi leikin piirteitä. Luonnollisesti myös

pelillisuus on osa leikillisyyttä – leikki on olennainen osa ihmiskunnan toimintaa kaikissa kulttuureissa ympäri maapalloa.

Leikillisten oppimateriaalien vahvuutena on niiden kyky edistää oppimista. Monipuolisella oppimateriaalien hyödyntämisellä jokainen lapsi voi löytää itselleen mieluisen tavan tutkia ja oppia uutta – tietyn aiheen monipuolinen opettelu materiaalien ja aktiviteettien avulla saattaa vahvistaa vahvojen käsitte-rakenteiden syntymistä sekä edistää lapsen oppimista.

Läpikäymissäni lähdemateriaaleissa ei varsinaisesti nostettu esiin leikillisen oppimisen huonoja puolia. Sen sijaan mainittiin leikillisyyden selkeimpänä heikkoutena se, miten nuorten ja aikuisten leikkiminen on hyvin marginaalista. Aikuisen leikkimistä aliarvostetaan ja leikkiä ei nähdä oleellisena osana ihmiskulttuuria toisin kuin aikuismaista ja ammattimaista toimintaa. Koulu-työssä pidemmälle edetessä leikkimistä on yhä vähemmän – mitä korkeam-masta luokka-asteesta on kyse, sitä kaavamaisempaa ja tylsempää luokkahuo-neen opetus on. Pelinomaiset käytännöt voisivatkin olla ratkaisu ihmisten leikillisyyden puuttumiseen.

On syytä pitää mielessä, että pelillistäminen ja leikillistäminen ovat opetuskeinoja, mutta silti on hyvä välillä opettaa myös käyttäen muita vaihto-ehtoisia opetuskeinoja. Esimerkiksi Angry Birds Playground -brändin hyödyn-täminen leikillisen oppimisen apuvälineenä saattaa tuntua turhan kaupalliselta, vaikka miten hyviä opetusmenetelmiä se tarjoaisikin (vrt. kohta 5.2).

Jatkossa olisi hyvä tehdä nykyistä kriittisempiä tutkimuksia siitä, miten pelillisuus vaikuttaa oppimistuloksiin. Tässä kirjallisuuskatsauksessa käytetyt lähteet olivat turhankin paljon pelillisyyden etuja hehkuttavia – esimerkiksi valtakunnan suurimmissa viestimissä nostetaan aika-ajoittain kritiikkiä koulu-maailman digiloikkaa ja opetuksessa käytettäviä sähköisiä laitteita kohtaan.

Leikillisyyteen liittyen puolestaan olisi suotavaa tehdä tieteellinen tutkimus siitä, miten opetuksen leikillistäminen toimii vanhemmilla oppilailla – esimer-kiksi murrosikäisillä yläkoululaisilla. Tästä aiheesta itseni olisikin hyvä tutkia viimeistään maisteritutkielman yhteydessä.

## Viiteluettelo

- Rachel A. Gordon. 2012. *Playful learning in early childhood*. The Illinois Report.
- Vilhelmiina Harju ja Jari Multisilta. 2014. Leikilliset oppimateriaalit innostavat oppimaan. Teoksessa Hannele Niemi ja Jari Multisilta (toim.), *Rajaton luokkahuone*. PS-kustannus, 270–284.
- J. Tuomas Harviainen ja Mikko Meriläinen. 2013. 1. Johdanto. Teoksessa J. Tuomas Harviainen, Mikko Meriläinen ja Tommi Tossavainen (toim.). 2013. *Pelikasvattajan käsikirja*. Pelikasvattajien verkosto. 7–9.
- Marjaana Kangas. 2010. *The School of The FuTure. Theoretical and Pedagogical Approaches for Creative and Playful Learning Environments*. Academic dissertation, Faculty of Education, University of Lapland.

- Marjaana Kangas. 2014. Leikillisyyttä peliin. Näkökulmia leikillisyyteen ja leikilliseen oppimiseen. Teoksessa Leena Krokfors, Marjaana Kangas ja Kaisa Kopisto (toim.), *Oppiminen pelissä. Pelit, pelillisuus ja leikillisuus opetuksessa*. Vastapaino, 73–92.
- Marjaana Kangas, Olli Vesterinen ja Leena Krokfors. 2014. Oppimispelit lapsen maailman, pelitutkimuksen ja osallistavan pedagogiikan risteyskohdassa. Teoksessa Leena Krokfors, Marjaana Kangas ja Kaisa Kopisto (toim.), *Oppiminen pelissä. Pelit, pelillisuus ja leikillisuus opetuksessa*. Vastapaino, 15–22.
- Fengfeng Ke. 2009. A qualitative meta-analysis of computer games as learning tools. In: Richard E. Ferdig (ed.), *Handbook of Research on Effective Electronic Gaming in Education*. Information Science Reference, 1–32.
- Harri Ketamo. 2014. Opettamalla oppii. Pelit osana koulutyöskentelyä. Teoksessa Hannele Niemi ja Jari Multisilta (toim.), *Rajaton luokkahuone*. PS-kustannus, 253–269.
- Kristian Kiili. 2005. Digital game-based learning: Towards an experiential gaming model. *Internet and Higher Education* 8, 13–24.
- Kristian Kiili, Pauliina Tuomi ja Arttu Perttula. 2012. Exerbraining for schools: Combining body and brain training. *Procedia Computer Science* 15, 2012, 163–173.
- Kristian Kiili, Pauliina Tuomi, Arttu Perttula ja Carita Kiili. 2014. Peleillä liikettä, luovuutta ja yhteisöllisyyttä koulupäivään. Teoksessa Hannele Niemi ja Jari Multisilta (toim.), *Rajaton luokkahuone*. PS-kustannus, 238–252.
- Jere Linnanen. 2014. Mistä on oppimispeli tehty? Omaa tietä etsimässä –oppimispeli ja sen tausta. Teoksessa Leena Krokfors, Marjaana Kangas ja Kaisa Kopisto (toim.), *Oppiminen pelissä. Pelit, pelillisuus ja leikillisuus opetuksessa*. Vastapaino, 276–282.
- Veli-Pekka Lifländer. 1989. *Tietokoneavusteisen opetuksen kehittäminen*. Helsingin kauppakorkeakoulun julkaisuja D-112.
- Tony Manninen. 2011. Pelien mahdollisuudet ja haasteet oppimisessa. Teoksessa Kimmo Oksanen, Birgitta Mannila ja Raija Hämäläinen (toim.), *Game Bridge. Kohti ammatillisia avaintaitoja*. Jyväskylän yliopisto, Koulutuksen tutkimuslaitos, 13–20.
- Elina Noppari. 2014. *Mobiilimuksut. Lasten ja nuorten mediaympäristön muutos, osa 3*. Journalismin, viestinnän ja median tutkimuskeskus COMET.
- Daniel Spikol and Marcelo Milrad. 2008. Physical activities and playful learning using mobile games. *Research and Practice in Technology Enhanced Learning* 3, 3, 275–295.
- Hiller A. Spires. 2008. 21st century skills and serious games: preparing the N generation. In: Leonard A. Annetta (ed.), *Serious Educational Games: From Theory to Practise*. Sense Publishers, 13–23.



- Tauno Tertsunen. 1999. *Toimintatutkimus tietokoneavusteisten opetusohjelmien hyödynnettävyydestä ammatillisessa koulutuksessa sähköalalla*. Syventävien opintojen tutkielma, Kasvatustieteen laitos, Helsingin yliopisto.
- Olli Vesterinen ja Jarkko Mylläri. 2014. Peleistä pelillisyyteen. Teoksessa Leena Krokfors, Marjaana Kangas ja Kaisa Kopisto (toim.), *Oppiminen pelissä. Pelit, pelillisuus ja leikillisuus opetuksessa*. Vastapaino, 56–66.